NASA/CR-1999-208973

A User's Manual for ROTTILT Solver: Tiltrotor Fountain Flow Field Prediction

Hormoz Tadghighi
The Boeing Company, Mesa, Arizona

R. Ganesh Rajagopalan
Iowa State University, Ames, Iowa

January 1999

NASA/CR-1999-208973

A User's Manual for ROTTILT Solver: Tiltrotor Fountain Flow Field Prediction

Hormoz Tadghighi
The Boeing Company, Mesa, Arizona

R. Ganesh Rajagopalan
Iowa State University, Ames, Iowa

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-20096

January 1999

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Contents

1 **Introduction** 1

2 **Flow Field Solution** 2

3 **Rotor Modeling** 3

3.1 Coordinates of the computational domain . . . 3

3.2 Rotor based Cartesian system . . . 3

3.3 Rotor based cylindrical polar system . . . 5

3.4 Coordinate system for blade deflection . . . 5

3.5 Rotor discretization . . . 5

3.6 Calculation of rotor forces . . . 6

4 **ROTTILT Program Architecture** 8

5 **Aircraft Body Components Section Cuts Generation** 8

6 **Grid Generation** 8

6.1 ROTTILT Internal Grid Generation and Quality assessment . . . 9

7 **Namelist File CNTRLPARMS** 9

7.1 Rotor Related inputs . . . 15

7.1.1 First Rotor Dimensions and Location . . . 15

7.1.2 Second Rotor . . . 16

7.2 Inputs Related to Other Bodies . . . 16

7.2.1 Wing . . . 16

7.2.2 First Nacelle . . . 17

7.2.3 Second Nacelle . . . 19

7.2.4 Fuselage . . . 19

7.3 Bounding Box Adjustment Notation . . . 20

8 **Grid Layering Information** 21

9 **Coupling of WOPWOP/TIN2 and ROTTILT** 24

10 **Validations** 24

11 **References** 31

12 Appendices **32**

A Sample Input File **33**

B ROTTILT Input Geometry Scale Down or Up Program **48**

C ROTTILT Grid Conversion into a 2D Gnuplot's Format **51**

D Geometry Section Cuts Generation Program **53**

E ROTTILT-WOPWOP-TIN2 Hi-Res Airloads and Inflow Velocities Conversions **57**

F Hess Write Format **68**

G 2D Gnuplot Plotting Routine **69**

H ROTTILT Input File for the XV15 Full Span Aircraft **74**

I Hess Format Sample Data **77**

List of Figures

1 Schematic of Fountain Flow Recirculation 1

2 Schematic of Rotor based Cartesian System. 3

3 A Schematic of Rotor Based Cylindrical System 4

4 Schematic of Rotor based Cartesian System. 4

5 A Schematic of Rotor Based Cylindrical System 6

6 A Schematic of Aerodynamic Forces on the Airfoil Section at ”s” 7

7 ROTTILT Solver Coupled with the Rotor(s) Trim Loop Flow Chart 11

8 ROTTILT Input Geometry Preparation Process 12

9 3D View of the Section Cuts for the Semi-Span XV15 Aircraft 12

10 ROTTILT Internal Grid Generation Process 12

11 XYZ Plane View of the Computational Grid of a Full-Span XV15 Aircraft 13

12 XV15 Semi-Span Configuration in ROTTILT Computational Grid Domain 14

13 Three-block grid specification. 21

14 ROTTILT Coupling with WOPWOP and TIN2 Flow Chart 24

15 XV15 Isolated Hover Performance 25

16 Numerical Simulation of Full Span XV15 Configuration in Hover - ROTTILT Velocity Trace 26

17 Numerical Simulation of Full Span XV15 Configuration in Hover - ROTTILT Velocity Vectors 27

18 Azimuthal Variation of Rotor Mean Inflow for Full Span XV15 in Hover 28

19 Radial Distribution of Local Inflow Velocity for Full Span XV15 in Hover - ROTTILT Solution 29

20 Comparison of Rotor Mean Inflow Velocity for a 15 percent Scaled JVX Rotor 29

21 Hover Acoustic Pressures for XV15 Aircraft: Along its Longitudinal Axis 30

22 Global coordinates and boundary conditions. 39

List of Tables

1 List of Namelist Parameters in CNTRLPARMS 10

2 List of Namelist Parameters in AdjustBox 15

3 List of Namelist Parameters in Layer Control 23

4 A1: Input/Output Print Control Parameters 36

5 A2: Flow Field Print Control Flags 37

6 A3: Solid Body Boundary Condition Control Flags 37

7 A4: Geometry and Flow Print Control Flags 38

8 A5: Flow Numerical Iteration Control Variables 38

9 A6: Numerical Relaxation Control Variables 39

10 A7: Flowfield Domain Boundary Condition Variables 40

11 A8: Flowfield Numerical Solution Directive Control Flags 40

12 A9: Computational Domain BC Setting Flags 41

13 A10: Numerical Relaxation Control Variables 41

14 A11: Rotor Airfoils Configuration Flags 42

15 A12: Flowfield Solution Directive Monitoring Variables 43

16 A13: Number of Rotor Configuration Control 43

17 A14: Rotor Airfoil Identification Command 43

18 A15: Rotor Blade Characteristics 44

19 A16: Rotor Configuration 44

20 A17: Rotor Blade Aerodynamic Control Point Setting 44

21 A18: Rotor Blade Geometry and Operating Controls 45

22 A19: Rotor Blade Flap and Pitich Motions 45

23 A20: Rotor Blade Geometry Definitions 46

24 A21: Rotor Aifoils Distribution Definitions 46

25 A22: Rotor Blade Airfoil Radial Distribution Control 46

26 A23: Rotor C81 Table File(s) Designation 47

27 A24: Aircraft Components Declaration 47

28 A25: Aircraft Trim Conditions 47

Abstract

A CFD solver has been developed to provide the time averaged details of the fountain flow typical for tiltrotor aircraft in hover. This Navier-Stokes solver, designated as ROTTILT, assumes the three-dimensional fountain flowfield to be steady and incompressible. The details of the theoretical background are described fully in this manual. In order to enable the rotor trim solution in the presence of tiltrotor aircraft components such as wing, nacelle, and fuselage, the solver is coupled with a newly developed set of dynamic trim routines which are highly efficient in CPU and suitable for CFD analysis. The Cartesian computational grid technique utilized in ROTTILT provides the user with a unique capability for insertion or elimination of any components of the bodies considered for the tiltrotor aircraft configuration by the user. Their presence in the flow field domain can be controlled by a simple logical switch in the input file. Flow field associated with either a semi or full-span configuration can be computed through user options in the ROTTILT input file. Full details associated with the numerical solution implemented in the solver are presented along with the assumptions. a brief description pertaining to the preparation of input surface mesh topology is provided with the listing for all the preprocessor programs in the appendices. Detailed definition of all the input variables with their default values listed in the main input files are described with reference to the V22 aircraft. Limited validation results for the V22 aircraft in hover obtained from the coupled ROTTILT/WOPWOP program is furnished for completion.

A pre-processor program based on the graphics package GNU-PLOT3D is also provided to visualize the computational grid. Sample input files for the V22 aircraft containing the geometry coordinates associated with the wing, nacelle, and fuselage components are also included to enable the reproduction of the solutions given in this report.

Nomenclature

b	= number of blades
c	= rotor blade chord
C_l,C_d	= sectional lift and drag coefficients, respectively
C_p	= pressure coefficient = $\frac{p-p_\infty}{q_\infty}$
D'	= drag force per unit span at a section
$\hat{e}_n, \hat{e}_\phi, \hat{e}_s$	= unit vectors in the (n, ϕ, s) system
$\hat{e}_r, \hat{e}_\phi, \hat{e}_z$	= unit vectors in the cylindrical coordinate system
$\vec{f}$	= resultant aerodynamic force on the blade, in the (n, ϕ, s) system
$\vec{F}$	= resultant aerodynamic force on the blade, in the (X, Y, Z) system
$\hat{I}$,$\hat{J}$,$\hat{K}$	= unit vectors in the $(X$,Y,$Z)$ system
L'	= lift force per unit span at a section
$\mathbf{M_1}$,$\mathbf{M_2}$,$\mathbf{M_3}$	= transformation matrices
n, ϕ, s	= coordinate system attached to the blade
p	= static pressure
q_∞	= free-stream dynamic pressure
R	= radius of the rotor
$\vec{\mathbf{R}}$	= position vector of a point on the blade span, in the (r, ϕ, z) system
s	= generalized coordinate measured along the blade span
S_X, S_Y, S_Z	= source terms in the discretized momentum equation
S'_X, S'_Y, S'_Z	= source terms in the momentum equation
$\vec{V}_{rel}$	= flow velocity relative to the blade
V_∞	= freestream velocity
$\vec{V}$	= absolute velocity of the flow with respect to the global coordinates
X, Y, Z	= reference frame attached to the computational domain
α	= blade angle of attack
α_D	= tip-path-plane angle of attack
δ	= blade deflection out of the rotor plane
ξ, η, ζ	= rotor-based Cartesian coordinates
θ_r	= blade pitch at the root
θ_s	= blade pitch at any section
θ_{tw}	= linear rate of twist
μ	= fluid viscosity
ρ	= fluid density
ψ	= azimuth angle
$\vec{\Omega}$	= rotational velocity of the rotor
$\vec{\zeta}$	= vorticity vector = $\nabla \times \vec{V}$
$\lVert\vec{\zeta}\rVert$	= magnitude of vorticity = $\sqrt{\zeta_X^2 + \zeta_Y^2 + \zeta_Z^2}$
ζ_X, ζ_Y, ζ_Z	= components of $\vec{\zeta}$

1 Introduction

Air traffic congestion at major airports throughout the world is quickly reaching the saturation point. The economic and political difficulties of constructing new major airports in heavily populated areas are enormous. Smaller commuter aircraft account for approximately 30 percent of airport usage while carrying only 5 percent of the passengers. A civil tiltrotor transport, operating in a National Airspace System tailored to permit vertiport access independent of airport control, would allow a significant increase in airport passenger movements. This would also minimize the requirement for new or large expansions to existing airports. The high noise levels generated by the tiltrotors and particularly during long periods in a terminal area could raise the ire community surrounding a vertiport. For tiltrotor transport to be a viable option the flow field and noise associated with terminal area operations must be understood.

The tiltrotor flow field in hover has been observed experimentally to possess many interesting features [Ref. 1]. Because the proprotor is nominally one wing chord length above the wing, the flow fields induced by the wing and the rotor are closely coupled. The flow field associated with the wing is largely unsteady and turbulent, and separated beneath the wing. As shown in Figure [1], the inboard-moving spanwise flow on the upper surface from both wing panels meets at the vehicle center-line and is redirected upward, and then downward through the rotors creating a recirculation pattern referred to as fountain flow.

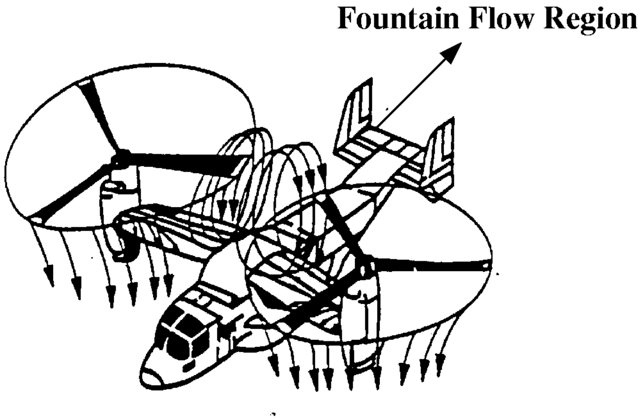

Figure 1: Schematic of Fountain Flow Recirculation

A CFD solver developed by the second author designated as ROTTILT solver [2] utilizes an efficient numerical technique using Navier-Stokes equations to simulate the flowfield associated with the tiltrotor aircraft in hover. The current version of the ROTTILT is equipped with an automated non-body fitted computational grid generation capability. The presence of the fountain flow for the titlrotor in hover is significant in contributing to the proprotor unsteady airloads, download or vertical drag, and its impulsive noise characteristics in hover.

The numerical prediction of all the flow features associated with the fountain flow still lies beyond the state-of-the-art. However, to render the problem tractable, the rotor is simplified in our analysis (see [Ref. 2]). A non-body conforming grid topology coupled with the ROTTILT solver offers a unique flexibility by accommodating a complex geometry such as semi-span/full-span tiltrotor configuration using a single grid block topology without sacrificing the accuracy of the solution sought. Using the grid control capabilities in the ROTTILT solver, we are able to employ a fine grid near the rotor disc plane. This preserves the accuracy of the numerical solution of the complex flow field around the rotor blade which is crucial for both performance and acoustic analysis. A non-body fitted computational grid is used for the wing/fuselage/nacelle to enhance the run time efficiency and hence reduce the dynamic memory requirement of the solver, but it also reduces the accuracy of the flow field around the bodies (i.e., wing, fuselage, nacelle) to a lower order. Since our primary goal of this case is not to perform accurate download calculations, the accuracy of the analysis in capturing the fountain flow effects is not degraded.

The aim of this user's manual is to provide instructions on utilizing the ROTTILT solver for applications to the tiltrotor aircraft configuration in hover. There are several preprocessors developed as stand-alone programs. Their utilization together with a public domain graphic package GNUPLOT is used to generate an accurate surface mesh which is required as input to the ROTTILT solver. Listings of these programs are given in the appendices. Step-by-Step procedure on input preparation to the ROTTILT for computational grid as well as ROTTILT's general input files are documented in this report. To decrease the complexity of the procedure for running the solver accurately, schematic flow charts are provided as a guide to generate input files. For completeness, formulations used in the ROTTILT are presented in full. However, this user manual is intended for the users who are familiar with CFD flow analyses pertaining to rotor-body flow interferences. Understanding the steps associated with the ROTTILT solver is vital for grid generation and input file preparation where many control flags are set to default values and should not be changed without a full understanding of the changes. Program listings of the post-processors developed for coupling of the ROTTILT with WOPWOP [Ref. 3] program are also included in the appendices of this report. Since the rotor mean downwash velocities are employed for turbulence ingestion noise prediction, a control flag for extracting the rotor inflow information at user specified planes is also added to the solver. As a basis for the validation of the solver, the results of the computed flow field and the associated aerodynamic and acoustic characteristics for the V22 and XV15 aircrafts respectively are presented.

2 Flow Field Solution

With the assumptions that the fluid density and viscosity are constant, the governing equations are based on steady, incompressible, laminar Navier-Stokes formulations. In the Cartesian coordinates system, the governing equations (i.e. continuity and momentum equations) can be written as:

Continuity:

$$\frac{\partial u}{\partial X} + \frac{\partial v}{\partial Y} + \frac{\partial w}{\partial Z} = 0 \tag{1}$$

X momentum :

$$\rho \left(u\frac{\partial u}{\partial X} + v\frac{\partial u}{\partial Y} + w\frac{\partial u}{\partial Z} \right) = \mu \left(\frac{\partial^2 u}{\partial X^2} + \frac{\partial^2 u}{\partial Y^2} + \frac{\partial^2 u}{\partial Z^2} \right) - \frac{\partial p}{\partial X} + S'_X \tag{2}$$

Y momentum :

$$\rho \left(u\frac{\partial v}{\partial X} + v\frac{\partial v}{\partial Y} + w\frac{\partial v}{\partial Z} \right) = \mu \left(\frac{\partial^2 v}{\partial X^2} + \frac{\partial^2 v}{\partial Y^2} + \frac{\partial^2 v}{\partial Z^2} \right) - \frac{\partial p}{\partial Y} + S'_Y \tag{3}$$

Z momentum :

$$\rho \left(u\frac{\partial w}{\partial X} + v\frac{\partial w}{\partial Y} + w\frac{\partial w}{\partial Z} \right) = \mu \left(\frac{\partial^2 w}{\partial X^2} + \frac{\partial^2 w}{\partial Y^2} + \frac{\partial^2 w}{\partial Z^2} \right) - \frac{\partial p}{\partial Z} + S'_Z \tag{4}$$

The above equations are solved using a finite-volume based method known as SIMPLER. Detailed information regarding the the principles of SIMPLER technique are given in Ref. 4.

The source terms S'_X, S'_Y, S'_Z added to the momentum equations are due to the rotor induced forces per unit volume acting at the cells which are encompassing the rotor plane. It is through these terms that the rotor's influence is introduced into the flow field. In effect, the rotor is represented as a a distribution of momentum sources acting in the rotor designated computational plane.

3 Rotor Modeling

In determining where the rotor's influence on the flow field is felt, the locations of the rotor physical plane in conjunction with the corresponding momentum equation source terms should be determined accurately. Therefore, the description of the rotor geometry is required (i.e. radius, chord, hub radius, etc.). To model the rotor in a Cartesian computational grid topology, four coordinate systems are employed. A brief description of each coordinate system as well as expressions for mutual transformations are presented in the following sections.

3.1 Coordinates of the computational domain

The governing equations are solved in the global (X, Y, Z) Cartesian coordinate system. $\hat{I}$, $\hat{J}$ and $\hat{K}$ are the unit vectors in the coordinate system. The center of the rotor is at (X_c, Y_c, Z_c) with respect to this system and its axis of rotation is along the vector $\vec{\Omega}$ where

$$\vec{\Omega} = \Omega_1 \hat{I} + \Omega_2 \hat{J} + \Omega_3 \hat{K} \tag{5}$$

and $|\vec{\Omega}| = \Omega$, the rotational speed in radians per second.

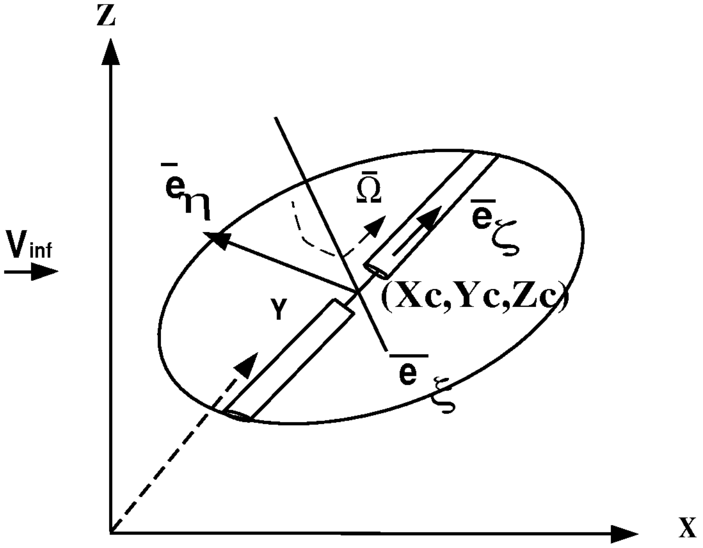

Figure 2: Schematic of Rotor based Cartesian System.

3.2 Rotor based Cartesian system

It is convenient to have the computational coordinates in the direction parallel and normal to the freestream velocity. The rotor blade orientation is arbitrary with reference to the freestream. Therefore a Cartesian coordinate system (ξ, η, ζ) which has its origin at the center of the rotor and the ξ axis in the direction opposite to the rotational velocity $\vec{\Omega}$ has been defined. As shown in Fig. 2, the ξ axis is perpendicular to the plane of rotation while the η and ζ axes lie in the plane. *Euler angle* method is employed to establish a relation between this system and the computational coordinate system. This results in an orthogonal transformation. Using this transformation which includes the shift of origin, the transformation from the rotor-based to the computational coordinates can be written as

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} \cos B & \sin A \sin B & -\cos A \sin B \\ 0 & \cos A & \sin A \\ \sin B \sin A & -\sin A \cos B & \cos A \cos B \end{bmatrix} \begin{bmatrix} X - X_c \\ Y - Y_c \\ Z - Z_c \end{bmatrix}$$

$$= \mathbf{M_1} \begin{bmatrix} X - X_c \\ Y - Y_c \\ Z - Z_c \end{bmatrix} \qquad (6a)$$

where A and B are two angles which describe the orientation of the rotor with respect to the computational coordinate system Ref [5].

A useful attribute associated with orthogonal transformations is that the inverse of the transformation matrix is its transpose. Thus we can write the inverse transformation from (ξ, η, ζ) to (X, Y, Z) as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} + \mathbf{M_1^T} \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} \qquad (6b)$$

The unit vectors in the two systems are also related by the matrix $\mathbf{M_1}$.

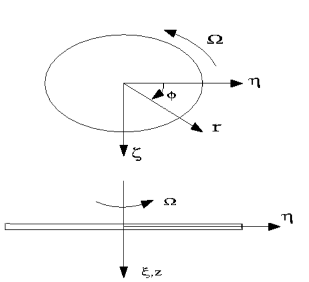

Figure 3: A Schematic of Rotor Based Cylindrical System

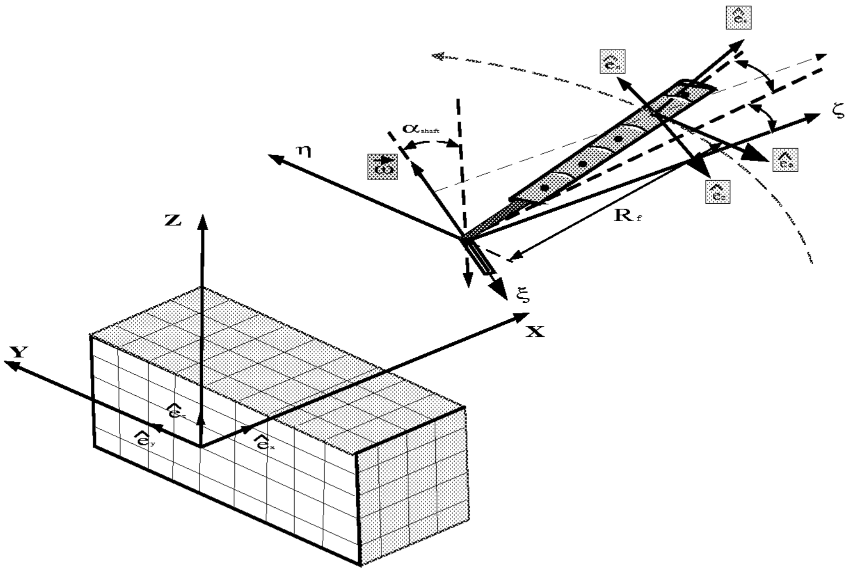

Figure 4: Schematic of Rotor based Cartesian System.

3.3 Rotor based cylindrical polar system

Further a cylindrical coordinate system (r, ϕ, z) as shown in Figs. 3 and 4 is defined which provides the necessary transformation from the Cartesian system to the blade coordinate system. The unit vectors in this system are related to those in the (ξ, η, ζ) system by the following matrix relation.

$$\begin{aligned}\begin{bmatrix} \hat{e}_r \\ \hat{e}_\phi \\ \hat{e}_z \end{bmatrix} &= \begin{bmatrix} 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{e}_\xi \\ \hat{e}_\eta \\ \hat{e}_\zeta \end{bmatrix} \\ &= \mathbf{M_2} \begin{bmatrix} \hat{e}_\xi \\ \hat{e}_\eta \\ \hat{e}_\zeta \end{bmatrix} . \end{aligned} \tag{7a}$$

The inverse relation can once again be obtained by taking the transpose of $\mathbf{M_2}$ such that

$$\begin{bmatrix} \hat{e}_\xi \\ \hat{e}_\eta \\ \hat{e}_\zeta \end{bmatrix} = \mathbf{M}_\mathbf{2}^{\mathrm{T}} \begin{bmatrix} \hat{e}_r \\ \hat{e}_\phi \\ \hat{e}_z \end{bmatrix} \tag{7b}$$

3.4 Coordinate system for blade deflection

To model elastic blade deformation, an additional coordinate system is required (n, ϕ, s), where s is in the spanwise direction of the blade (i.e., s is the location of the aerodynamic center of the airfoil sections). A line sketch in Fig. 5 depicts a curved blade. The direction $\hat{e}_\phi$ is the same as in the previous system and $\hat{e}_n$ is defined to complete the right handed system. Thus the (n, s) axes always lie in the r-z plane and, when $\delta = 0$, the n axis opposes z while the s axis coincides with r. The transformation between this and the cylindrical system can be written as

$$\begin{aligned}\begin{bmatrix} \hat{e}_n \\ \hat{e}_\phi \\ \hat{e}_s \end{bmatrix} &= \begin{bmatrix} \sin\delta & 0 & -\cos\delta \\ 0 & 1 & 0 \\ \cos\delta & 0 & \sin\delta \end{bmatrix} \begin{bmatrix} \hat{e}_r \\ \hat{e}_\phi \\ \hat{e}_z \end{bmatrix} \\ &= \mathbf{M_3} \begin{bmatrix} \hat{e}_r \\ \hat{e}_\phi \\ \hat{e}_z \end{bmatrix} \end{aligned} \tag{8a}$$

and inversely

$$\begin{bmatrix} \hat{e}_r \\ \hat{e}_\phi \\ \hat{e}_z \end{bmatrix} = \mathbf{M}_\mathbf{3}^{\mathrm{T}} \begin{bmatrix} \hat{e}_n \\ \hat{e}_\phi \\ \hat{e}_s \end{bmatrix} \tag{8b}$$

Given the distribution of the deflection along the blade span, the following equation of the curved blade can be easily derived

$$\vec{\mathbf{R}}(s) = \hat{e}_r \int_0^s \cos\delta(s)\ ds + \hat{e}_z \int_0^s \sin\delta(s)\ ds \tag{9}$$

3.5 Rotor discretization

The rotor blades are discretized into spanwise elements. Blade properties such as chord length, out of plane deflection, twist, thickness and the airfoil section characteristics at the control point of each element are assumed to be constant across the length of the element. The control points for the blade segments should prescribe a circular path which is aligned with the blade aerodynamic center. Therefore, it is important

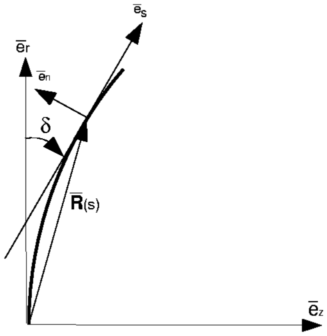

Figure 5: A Schematic of Rotor Based Cylindrical System

to locate the grid cells of the computational domain which are in the path of the rotor circular motion. Since the three-dimensional computational grid is oriented arbitrarily with respect to this circle, a general algorithm has been developed for this purpose. Details can be found in Ref. 6.

3.6 Calculation of rotor forces

Let the fluid velocity at any point s on the blade element at an angular position ϕ be

$$\vec{V} = u\hat{I} + v\hat{J} + w\hat{K}. \tag{10}$$

Using equations (6a), (7a) and (8a) the same can be written in the (n, ϕ, s) system as

$$\vec{V} = v_s\hat{e}_s + v_\phi\hat{e}_\phi + v_n\hat{e}_n \tag{11}$$

where

$$\begin{bmatrix} v_s \\ v_\phi \\ v_n \end{bmatrix} = \mathbf{M_3M_2M_1} \begin{bmatrix} u \\ v \\ w \end{bmatrix}.$$

The blade has a velocity due to its rotation which can be written in the (n, ϕ, s) system as

$$\vec{V_{bl}} = \mathbf{M_3}\left(\mathbf{M_2M_1}\vec{\Omega} \times \bar{\mathbf{R}}(s)\right) \tag{12}$$

where $\vec{\Omega}$ is defined in Equation (5) and $\vec{\mathbf{R}}(s)$ is the position vector of the point on the blade under consideration. Hence the flow velocity relative to the blade, $\vec{V}_{rel} = v'_s\hat{e}_s + v'_\phi\hat{e}_\phi + v'_n\hat{e}_n$ is given by

$$\begin{aligned} \vec{V}_{rel} &= \vec{V} - \vec{V}_{bl} \\ &= \mathbf{M_3M_2M_1}\vec{V} - \mathbf{M_3}\left(\mathbf{M_2M_1}\vec{\Omega} \times \vec{\mathbf{R}}(s)\right) \end{aligned} \tag{13}$$

In order to determine the aerodynamic forces on the airfoil section at s, we need only the component of $\vec{V}_{rel}$ in the plane normal to $\hat{e}_s$. The angle made by this component with the $\hat{e}_\phi$ direction is given by (see Fig. 6)

$$\beta = \arctan(-v'_n/v'_\phi) \tag{14}$$

If the blade has an angle of twist θ_s with respect to the plane of rotation then, from Fig. 6, the effective angle of attack seen by 2D airfoil section is

$$\alpha = \theta_s - \beta. \tag{15}$$

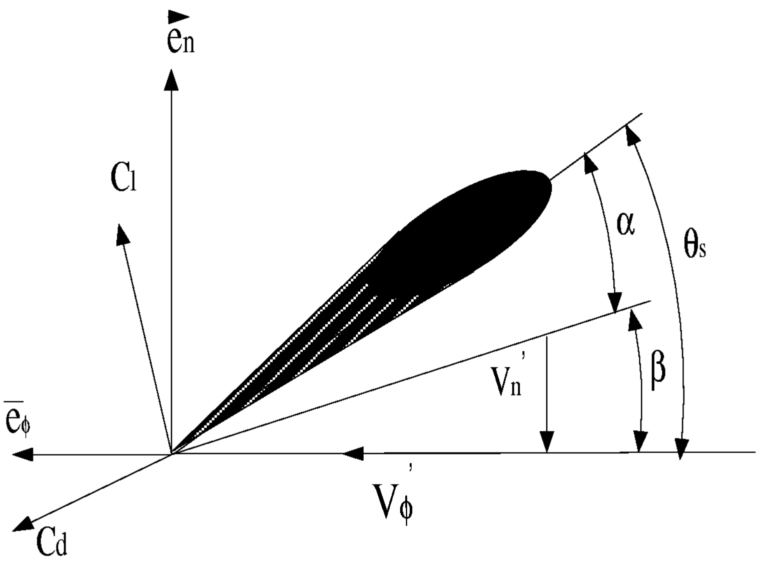

Figure 6: A Schematic of Aerodynamic Forces on the Airfoil Section at "s"

Knowing the angle of attack α and the components of the relative flow velocity experienced by the airfoil section, we can find the sectional aerodynamic force coefficients C_l and C_d from the classical 2D blade element formulations. The lift and drag forces on the blade element of length ds can then be found from

$$\mathrm{L}' = \frac{1}{2}\rho v'^2 \mathrm{C}_l c \; ds \tag{16a}$$

$$\mathrm{D}' = \frac{1}{2}\rho v'^2 \mathrm{C}_d c \; ds \tag{16b}$$

where c is the blade chord-length and $v'^2 = {v'_n}^2 + {v'_\phi}^2$. The lift and drag forces act perpendicular and parallel, respectively, to the relative velocity vector. Resolving these forces in the $\hat{e}_\phi$ and $\hat{e}_n$ directions, we have

$$f_n = \mathrm{L}' \cos\beta - \mathrm{D}' \sin\beta \tag{17a}$$

$$f_\phi = \mathrm{L}' \sin\beta + \mathrm{D}' \cos\beta. \tag{17b}$$

Also, since there are no aerodynamic forces along the span,

$$f_s = 0.$$

Thus the resultant aerodynamic force on the blade segment, $\vec{f} = (f_n, f_\phi, f_s)$, can be computed for the (n, ϕ, s) system. The corresponding force vector in the (X, Y, Z) system, $\vec{F}$, can be found by using the inverse transformation relations (8b), (7b) and (6b) as

$$\vec{F} = \mathbf{M_1}^{\mathrm{T}} \mathbf{M_2}^{\mathrm{T}} \mathbf{M_3}^{\mathrm{T}} \vec{f}. \tag{18}$$

The instantaneous force acting on the fluid element at the (s, ϕ) location is, then, $-\vec{F}$. Since the blade actually spends a finite fraction of its total revolution time passing through this control volume, the time averaged source terms $\vec{S} = (S_X, S_Y, S_Z)$ should be added to the discretized momentum conservation equations at the control volume which is the result of $-\vec{F}$ multiplied by its fractional time value, i.e.,

$$\vec{S} = \frac{b\,\Delta\phi}{2\pi}(-\vec{F}) \tag{19}$$

where b is the number of blades and $\Delta\phi$ is the angular distance through which the blade traverses in passing through the specific control volume.

4 ROTTILT Program Architecture

In order to eliminate any computational errors associated with the coupling of the ROTTILT solver with external rotor trim program such as CAMRAD, trim loop has been added to the solver. For the user's specified Ct/sigma, the collective pitch angle is iteratively perturbed. As shown in Fig.(7), a final flow solution for any given tiltrotor configuration is obtained when both the flowfield and rotor trim are converged simultaneuosly. The rotor trim condition is controlled by the user's input variables which are furnished in the rotor trim namelist section of solver main input file. Details regarding their definitions are given in the following sections. Integration of the rotor trim loop into the ROTTILT has increased the run CPU time of the solver by only 5 percent which is negligible compared with the overall run time of the solver. This is due to the implementation of the trim loop as a sub-iteration to the solver main flow field iteration loop, (see Fig. (7)).

5 Aircraft Body Components Section Cuts Generation

Conventially, the aircraft body geometry is assumed to be obtained by the user from a CADDY system (e.g. Boeing's UG system). A translator is required to convert the CADDY's design part into a surface mesh format such as HESS which is ascii and it is generally considered to be a standard CFD file format. A sample file containing the HESS format is given in Appendix (I). Using the HESS file containing the surface mesh geometry coordinates for the user specified components (i.e. fuselage, wing, and nacelle), a preprocessor SPLINE can be employed to generate the section cuts which are required as input to the ROTTILT grid generation program. As an option, SPLINE program provides equally the same number of graphic files for the user checks on the section cuts computed by the SPLINE program. Public domain graphic program GNUPLOT, which is 3D and interactive is then used to visualize the cuts before proceeding to the ROTTILT grid generation state. As a successful completion of the geometry manipulation process, one file per component is generated with the component's name and an extension ".dat" (e.g. Wing.dat, Fuselage.dat, etc). These files are ascii and hence they are not machine dependent which make them easily portable to any computer platform. Overall architecture of the process flow chart is shown in Fig. (8). Also, a typical 3D plot of the XV15 semi-span section cuts generated from this process coupled with the GNUPLOT graphic can be seen in Fig. 9. Using GNUPLOT graphic control commands the displayed aircraft configuration can be scrutinized for quality of the section cuts.

6 Grid Generation

The Grid generation program lays out the computational grid over a tiltrotor's various components. The user needs to specify the information related to the surface geometries of the bodies in a data file. Input file format for tiltrotor bodies such as Rotor, Wing, Nacelle and Fuselage components are fully described in this documentation. It important to mention that the input lines that begin with the "#" (Number Sign) as the first character indicates a comment statement (i.e. inactive input).

The input file provides the following control parameters which are required as input to the Grid Generation program:

- Executive control parameters for the grid generation
- Rotor and its Dimensions
- Rotor center location with respect to user specified common axis employed for all other components.
- Number of grid points to be generated on the rotor disk

- Existence of the aircraft components like Wing, Nacelle(s) and Fuselage.
- For each one of the existing bodies
  - Component name file containing the body points.
  - Number of grid cells required on the body (in I and J directions)
    The grid generation program will use the user specified cell density as a guidance to generate the actual computational grid.
  - The extent of the bounding box for each component of the aircraft.
- User specified grid layer file name containing the grid density specification.

6.1 ROTTILT Internal Grid Generation and Quality assessment

Since Computational grid generation is an integral part of the ROTTILT solver, it is advisable to ensure the quality of the grid in terms of its density and distribution per component following the process depicted in the Fig. (10). Therefore, with setting the flag "StpAftrGD" to "true", the computational grids generated as output for each aircraft body component are written separately into their respective part names appended with an extension of ".out" in a 2D format. Postprocessing program Conv.f can be used to convert the grid files into graphic postscript format which can then be visualized using "GOSTVIE", as shown in Fig. (11). Three view angles of the computational grid is provided to the user namely, XY, XZ, and YZ planes. If the grid distribution is not acceptable as a whole or partially, after adjusting the grid control parameters in the ".lay" file, a new set of grids can then be obtained through an iterative process with ROTTILT before initiating the solver. As the flow field solution around the rotor domain is of primary importance here, the user should ensure that the grid distribution in the rotor region is uniform as shown in Fig.(11). Abrute force increase in the grid density will naturally increases the CPU run time significantly. User experience with the solver is therefore essential in balancing the solution accuracy versus grid density for the ROTTILT solver. Furthermore, the ROTTILT generates an output grid file in FAST format, as shown in Fig. (12). The various grid block for each aircraft components with their density are clearly depicted. It is important to mention that the computational grid for each block overlaps the neighboring block thus creating regions of high density grid. With the user's experience, the fringe regions between the blocks can be tailored in such a fashion to improve the grid quality in the rotor domain without increasing the total number of grid points.

7 Namelist File CNTRLPARMS

Control parameters are provided as Namelist parameters and they begin with $CNTRLPARMS and ends with $END. The keywords $CNTRLPARMS and $END must start at the second character in the line.

In Table [1], the actual namelist parameters are specified between the $CNTRLPARMS and $END. Each one of the parameters need to be given on a separate line. If any of the parameters is not provided then the default values would be used.

In Table [2], the Bounding Box Adjustment may be done beyond the bounding box of all bodies in the I, J, and K directions. Setting to TRUE will allow the grid generation program to have no constraint on extending the boundaries of bounding boxes beyond the specified maximum and minimum of the box. For example, the grid over the wing should not extend beyond Y-min and hence AdjstByndYL should be FALSE.

Table 1: List of Namelist Parameters in CNTRLPARMS

Parameter Name	Value	Description(Meaning of TRUE)	Default
DBGlvl	Integer [1 to 9]	Print the values of fields	0
DBGpltLvl	Integer	Used in debugging	0
FrstNacl	TRUE/False	First Nacelle	False
ScndNacl	TRUE/False	Second Nacelle	False
WingBody	TRUE/False	Wing	False
Fuselage	TRUE/False	Fuselage	False
ScndRtr	TRUE/False	Second Rotor	False
FullSpan	TRUE/False	Full span	False
StpAftrGd	TRUE/False	Stop execution after Grid Generation	False
BothWings	TRUE/False	Both Wings exist	False
ExtndWing	TRUE/False (Useful when there is no fuselage)	Extend Wing to the center of the Fuselage	False

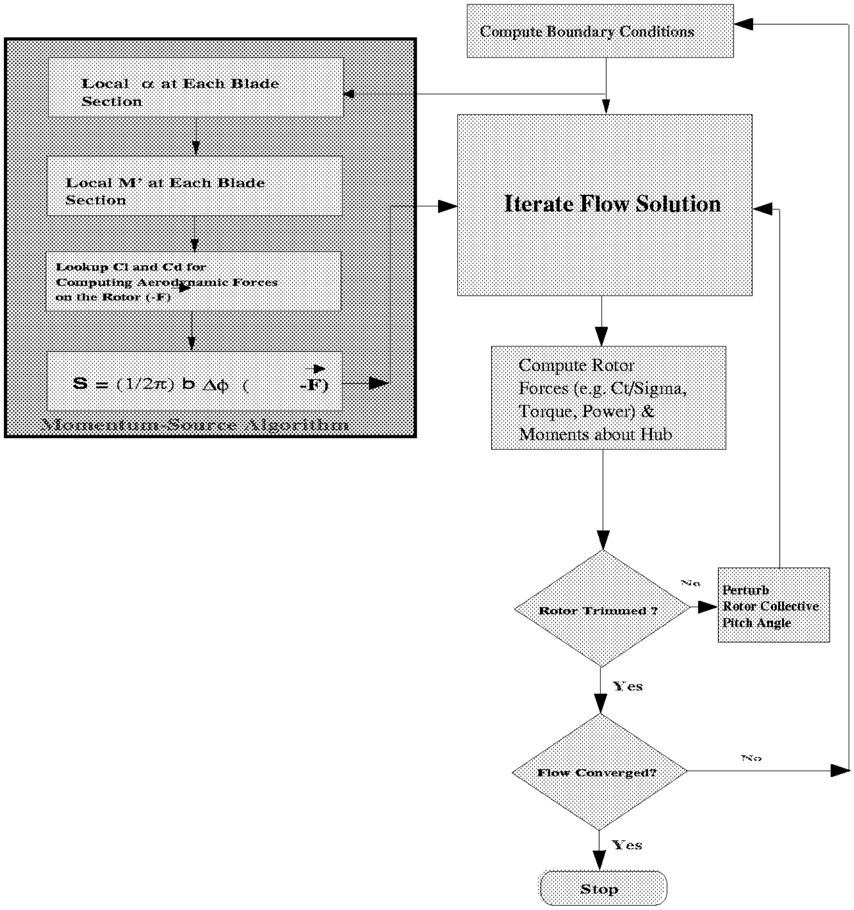
Compute Boundary Conditions
Local α at Each Blade Section
Local M' at Each Blade Section
Lookup Cl and Cd for Computing Aerodynamic Forces on the Rotor (-F)
S = (1/2π) b Δφ (-F)
Momentum-Source Algorithm
Iterate Flow Solution
Compute Rotor Forces (e.g. Ct/Sigma, Torque, Power) & Moments about Hub
Rotor Trimmed ?
No
Perturb Rotor Collective Pitch Angle
Yes
Flow Converged?
No
Yes
Stop

Figure 7: ROTTILT Solver Coupled with the Rotor(s) Trim Loop Flow Chart

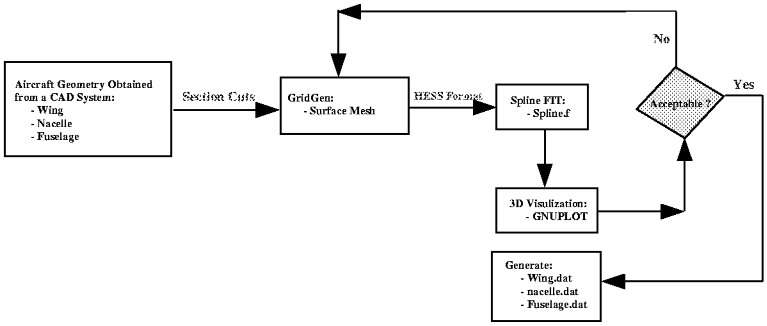

Figure 8: ROTTILT Input Geometry Preparation Process

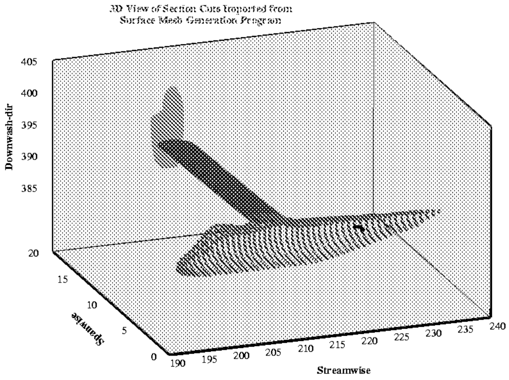

Figure 9: 3D View of the Section Cuts for the Semi-Span XV15 Aircraft

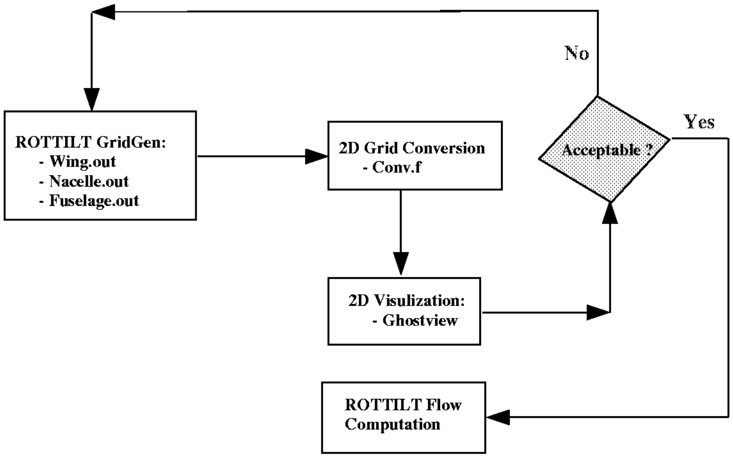

Figure 10: ROTTILT Internal Grid Generation Process

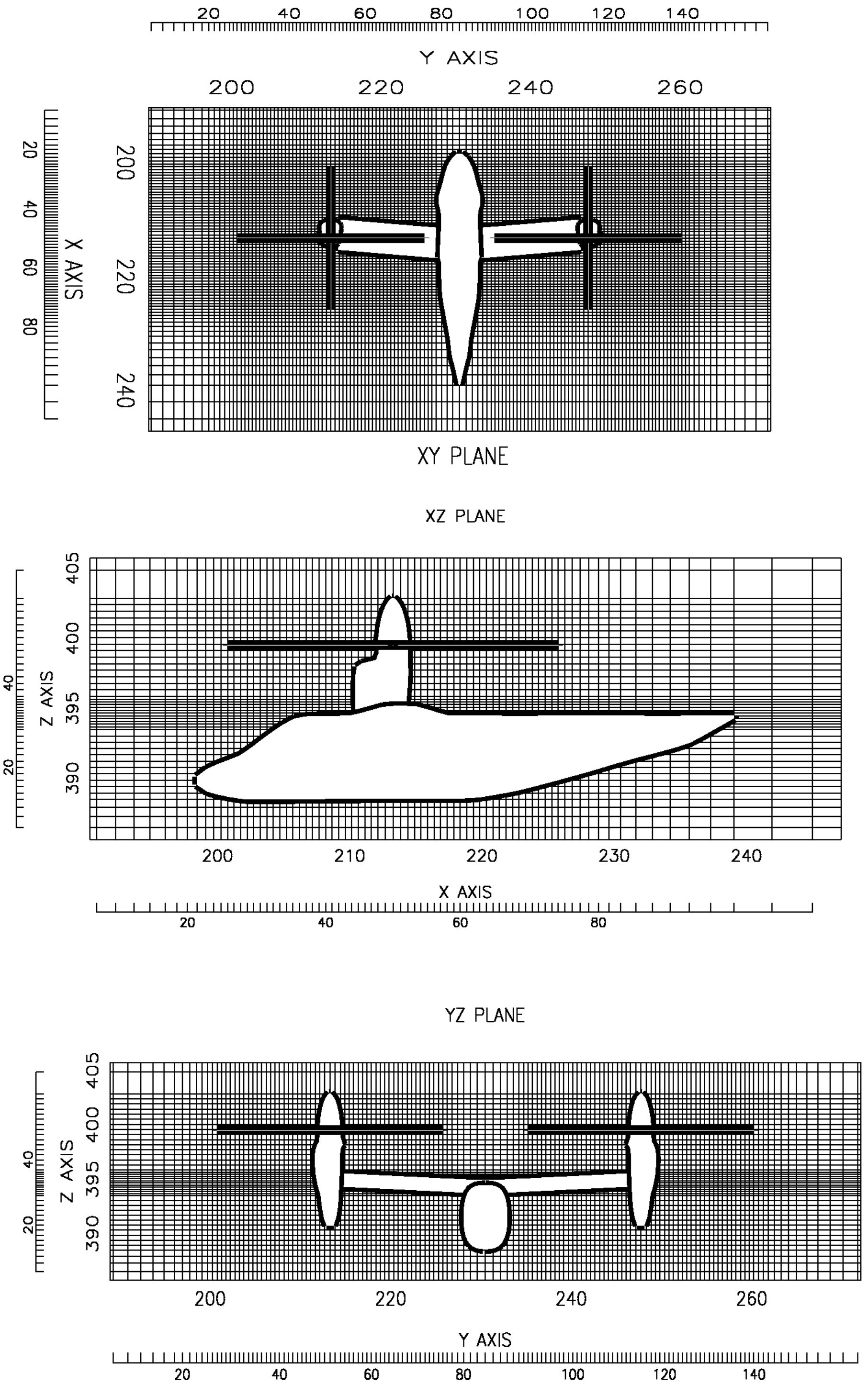

Figure 11: XYZ Plane View of the Computational Grid of a Full-Span XV15 Aircraft

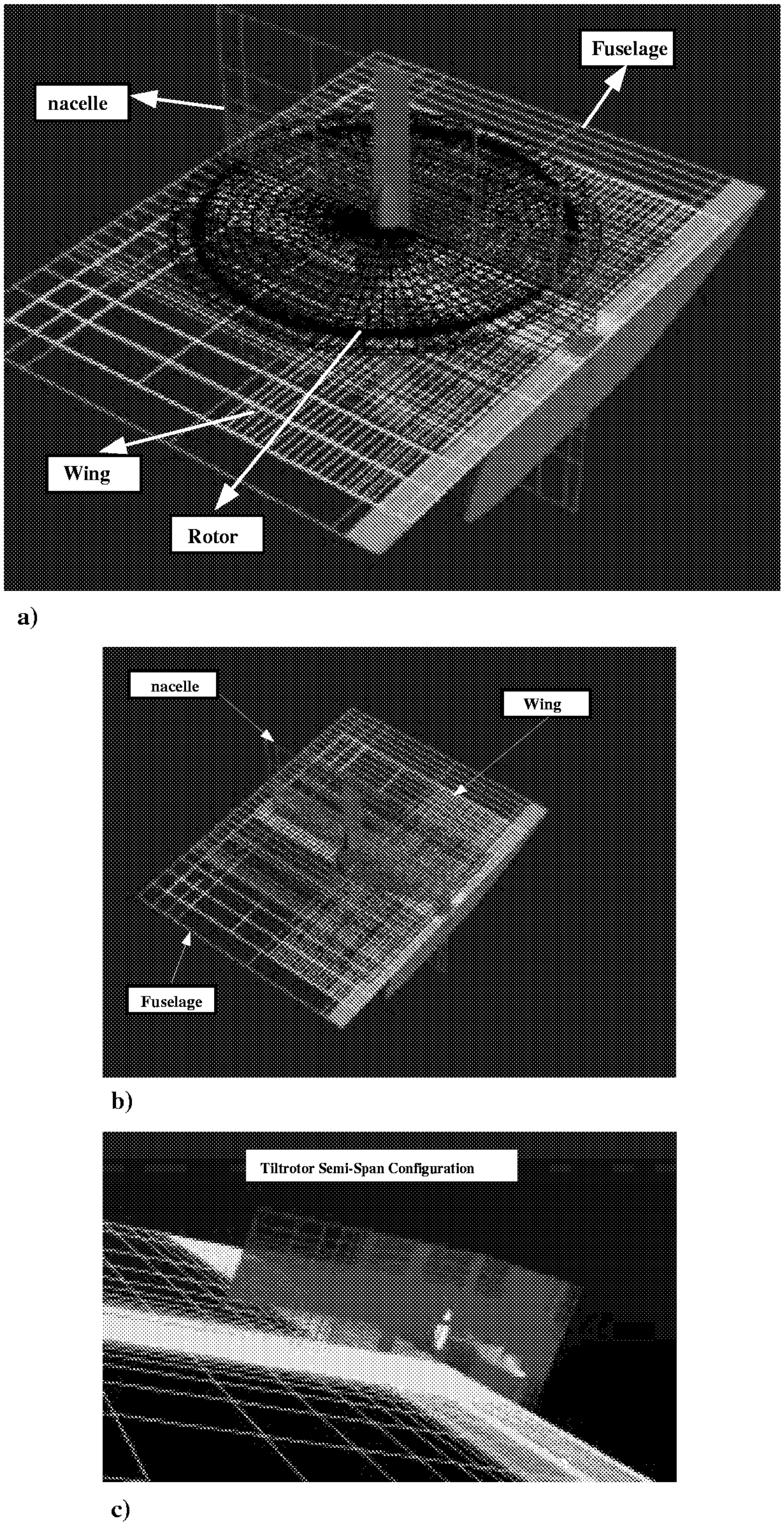

Figure 12: XV15 Semi-Span Configuration in ROTTILT Computational Grid Domain

Table 2: List of Namelist Parameters in AdjustBox

Parameter Name	Value	Description(Meaning of TRUE)	Default
AdjstByndXL	TRUE/False	Low value in X direction	TRUE
AdjstByndXH	TRUE/False	High value in X direction	TRUE
AdjstByndYL	TRUE/False	Low value in Y direction	TRUE
AdjstByndYH	TRUE/False	High value in Y direction	TRUE
AdjstByndZL	TRUE/False	Low value in Z direction	TRUE
AdjstByndZH	TRUE/False	High value in Z direction	TRUE

A typical example for the namelist file "CNTRLPARMS" is given below:
Example:

```
$CNTRLPARMS
   DBGlvl = 1
   DBGpltLvl = 1
   WingBody = .TRUE.
   FrstNacl = .TRUE.
   Fuselage = .TRUE.
   AdjstByndYL = .FALSE.
$END
```

7.1 Rotor Related inputs

In a multi rotor configuration, the rotor information is provided through the first rotor input parameters only. For example, in the case of a dual rotor configuration, the second rotor is simply treated as a mirror image to the first rotor. The following sections explain the details of the input specifications for the rotor.

7.1.1 First Rotor Dimensions and Location

Example:

```
#
# Rotor Radius, Half Thickness , Rotor Center (X,Y,Z) Coordinate
12.5, 0.3, 26.8478, 17.227, 18.400
#
```

Explanation:

Rotor Radius in feet
Half Thickness (.i.e half of the height of the computational box which is encompassing the rotor) is specified as a ratio of the Rotor Radius
(i.e actual Half thickness (half Z-width) by Rotor Radius)

This info is used only for grid generation (to specify # of grids in K dir)

X coordinate for Rotor Hub Center

Y coordinate for Rotor Hub Center

Z coordinate for Rotor Hub Center

First Rotor - No of Grids

The approximate number of grid points required on the First Rotor in I, J , K directions respectively need to be provided as follows :

Example:

```
# Rotor I-Grids, J-Grids and K-Grids
    30, 14, 2
#
```

First Rotor - Bounding Box Adjustment

The bounding box for the Rotor is initially set up by adding and subtracting the rotor radius from the three coordinates of the Rotor center. The size of this bounding box can be adjusted, using similar procedure and convention as the body bounding box adjustments explained previously.

Bounding Box Adjustments for rotor are specified as a ratio of the rotor diameter.

So if the user wants the bounding box to be

- extended then a Positive Ratio needs to be set
- reduced then a Negative Ratio needs to be set

7.1.2 Second Rotor

If the Second Rotor exists then all the above inputs specified for the First Rotor need to be given for the Second rotor also.

7.2 Inputs Related to Other Bodies

The other bodies that may be defined are Wing, Nacelles and Fuselage. This chapter explains the user specifications for the grid over these bodies.

7.2.1 Wing

If wing exists (WingBody = TRUE) then the wing data is given as follows

Wing - Body Points File

The Wing points are specified in a separate file and the name of the file is provided as follows

Example:

```
# Wing File Name (File Containing Wing Points)
’BdyPts/Wing.dat’
#
```

Wing - No of Grids

The approximate number of grid cells required on the Wing in I, J , K directions respectively need to be provided as follows:

Example:

```
#
# Number of I-Grids, J-Grids and K-Grids for Wing
    12, 12, 12
#
```

Wing - Bounding Box Adjustment

The Adjustment Ratios for the Bounding Box around the Wing follows the same notation and convention as for the bounding box adjustments for the body and rotor explained previously.

Example:

```
#
# Adjust Wing BBox (MinX, MaxX, MinY, MaxY, MinZ, MaxZ) by Ratios
    0.06, 0.18, -0.045136, -0.02639252, 0.45, 0.27
#
```

7.2.2 First Nacelle

If First Nacelle exists (FrstNacl = TRUE) then the First Nacelle data is given as follows

First Nacelle - Body Points File

The Nacelle points are specified in a separate file and the name of the file is specified as follows

Example:

```
#
# Nacelle File Name (File Containing Nacelle Points)
```

```
        'BdyPts/Nacelle.dat' (with "BdyPts" being the path)
#
```

First Nacelle - No of Grids

The approximate number of grid cells required on the First Nacelle in I, J , K directions respectively need to be provided as follows:

Example:

```
# Number of I-Grids, J-Grids and K-Grids for Nacelle
        1, 7, 14
#
```

First Nacelle - Bounding Box Adjustment

The Adjustment Ratios for the Bounding Box around the Nacelle are specified as per the notation explained in Section 7.3 ;

Example:

```
# Adjust Nacelle-1 BBox (MinX, MaxX, MinY, MaxY, MinZ, MaxZ)
# by Ratios (before Grid Gen. on Bodys):
    0.0, 0.0, 0.0, 0.0, -0.45, -0.13
```

7.2.3 Second Nacelle

If Second Nacelle exists (ScndNacl = TRUE) then the Second Nacelle data is given just like the First Nacelle.

7.2.4 Fuselage

If Fuselage exists (Fuselage = TRUE) then the Fuselage data is given as follows:

Fuselage - Body Points File

The Fuselage points are specified in a separate file and the name of file is specified as follows:

Example:

```
#
# Fuselage File Name (File Containing Fuselage Points),
# Translated Fuselage Pts. Out File Name
    'BdyPts/Fuselage.dat'
#
```

Fuselage - No of Grids

The approximate number of grid cells required on the Fuselage in I, J , K directions respectively need to be provided as follows:

Example:

```
# Number of I-Grids, J-Grids and K-Grids for Fuselage
    20, 5, 11
#
```

Fuselage - Bounding Box Adjustment

The Adjustment Ratios for the Bounding Box around the Fuselage are specified as per the notation explained in Section 7.3 ;

Example:

```
# Adjust Fuselage BBox (MinX, MaxX, MinY, MaxY, MinZ, MaxZ)
# by Ratios (before Grid Gen. on Bodys):
    0.15, 0.15, 0.0, 0.0, 0.0, -0.37
#
```

7.3 Bounding Box Adjustment Notation

The bounding box for a body is initially set based on the geometry of the body. A bounding box bounds the body completely. Adjusting the bounding box allows a clearance between the geometry and the surrounding box.

This Bounding Box may be adjusted using the input ratios provided by the user.
The adjustment in a direction is provided as a ratio of the total dimension in that direction.

Example:

```
# Adjust Rotor-1 BBox (MinX, MaxX, MinY, MaxY, MinZ, MaxZ) by Ratios
    0.15, 0.15, -0.5662888, 0.15, -0.82, 0.07
#
```

The above 6 entries correspond to Bounding Box Adjustment Ratio for the following:

X minimum

X maximum

Y minimum

Y maximum

Z minimum

Z maximum

In the above example, the original bounding box at the starting X location (X minimum) is enlarged by the size of 0.15 times the total dimension of the body. A negative ratio simply means that the original bounding box is reduced instead of being enlarged.

So if the user wants the bounding box to be

- extended, then a Positive Ratio needs to be set
- reduced, then a Negative Ratio needs to be set

8 Grid Layering Information

The grid specification of the total computational domain follows a **three-block** arrangement, which are designated as Block#1, Block#2, and Block#3, as given in Figure (13). Block#2 contains the rotor and the bodies, and its grid specification is given automatically by the program. Grids outside the rotor-body block up to the boundaries of the computational domain are specified by the user, by providing the layer information of Block#1 and Block#3 in the three coordinate directions. The grid generation program needs this layer information which is specified in a separate file.

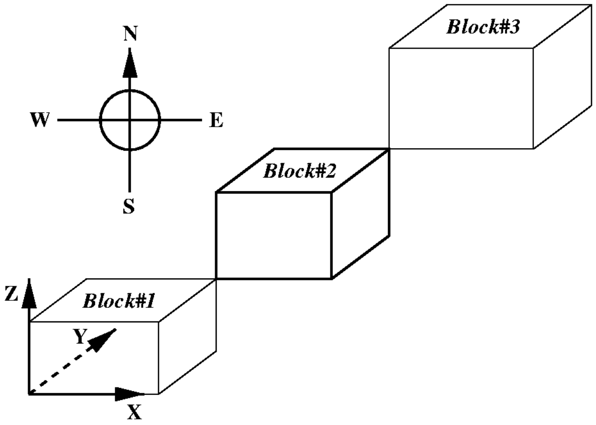

Figure 13: Three-block grid specification.

In the current input file, the name of the layer information file is specified as follows:

Example:

```
# Layer Information File
      'Layer.dat'
#
```

The layer information for Block#1 and Block#3 in the "Layer.dat" file must follow the following format:

Example:

```
# Block#1
#——————————————————
# Layer Definition for Grid before body (from South-West-Bottom to Body Start)
#——————————————————
#
# NO OF LAYERS (X-grid)
    2
#—————————————————————————
# LAYER# RLEN CLEN NCEL RATX
#—————————————————————————
      1        14.0     12.5     7     -1.65
      2         1.0     12.5     4     -1.05
#
```

```
# NO OF LAYERS (Y-grid)
   0
#————————————————————————————
# LAYER# RLEN CLEN NCEL RATX
#————————————————————————————
#
# NO OF LAYERS (Z-grid)
   4
#————————————————————————————
# LAYER# RLEN CLEN NCEL RATX
#————————————————————————————
      1       24.0    12.5    4     -1.3
      2       5.92    12.5    4     -1.4
      3         1.0    12.5    3     -1.9
      4         2.0    1.0      3     -1.2
#
# Block#3
#——————————————————————
# Definition for Grid after body (from Body End to North-East-Top)
#——————————————————————
#
# NO OF LAYERS (X-grid)
   2
#————————————————————————————
# LAYER # RLEN CLEN NCEL RATX
#————————————————————————————
      1         1.0    12.5    4    1.05
      2       14.0    12.5    7    1.65
#
# NO OF LAYERS (Y-grid)
   2
#————————————————————————————
# LAYER # RLEN CLEN NCEL RATX
#————————————————————————————
      1         1.0    12.5    -5    1.15
      2       14.0    12.5    -7    1.65
#
# NO OF LAYERS (Z-grid)
   2
#————————————————————————————
# LAYER # RLEN CLEN NCEL RATX
#————————————————————————————
      1       1.0    12.5      3    1.85
      2       9.0    12.5      4    1.6
#
```

The parameters in the layer information file are explained in the following table:

Table 3: List of Namelist Parameters in Layer Control

Parameter Name	Values Type	Description
NO OF LAYERS	Integer	Number of layer in a particular coordinate direction
LAYER	Integer	Layer number
RLEN	Real	Specified length
CLEN	Real	Characteristic length RLEN times CLEN gives the true length CLEN=1 means RLEN is the true length
NCEL	Integer	Number of grid cell Negative number means equally spaced grid
RATX	Real	Negative value means that the next cell width (CW_{i+1}) is smaller than the previous value (CW_i) and the ratio CW_i/CW_{i+1} = 1/ABS(RATX) Positive value means that the next cell width is greater than the previous value and the ratio CW_{i+1}/CW_i = ABS(RATX)

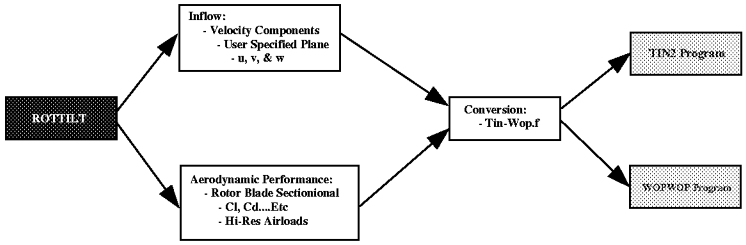

Figure 14: ROTTILT Coupling with WOPWOP and TIN2 Flow Chart

9 Coupling of WOPWOP/TIN2 and ROTTILT

The acoustic analysis of the tiltrotor in hover is performed using the acoustic prediction code WOPWOP [3]. Presently, the noise sources associated with the fountain flow effects are classified into the tonal noise (i.e deterministic) as well as the low frequency broadband noise (non-deterministic), as discussed in Ref [9]. Current version of the ROTTILT computed high resolution airloads and mean rotor inflow downwash which are employed as input to WOPWOP and TIN2 programs. No comparative studies have yet been made to determine their relative contributions to the overall noise in the presence of fountain flow effects. However, it is important to mention that an accurate low frequency noise computation resulting from the fountain flow is strongly dependent on the partial inflow turbulence induced by the flow recirculation. The prediction of the noise sources due to the inflow of turbulence is currently beyond the scope of the CFD solver presented here, but the utilization of a newly developed empirical formulation(s) based on well defined experimental data for the turbulence characteristics associated with this phenomena would be an attractive candidate for low frequency broadband noise analysis for tiltrotor in hover. Further, the TIN2 program inflow distortion matrix employs the rotor inflow information over the whole rotor plane which can be computed using ROTTILT solver. For WOPWOP, the high resolution airloads (e.g. per 0.5 degs) computed by ROTTILT solver are used for the noise analysis A postprocessing code WOPPOST (Appendix E) is used to convert the computed ROTTILT airloads into a suitable format for the WOPWOP. Compact version of the WOPWOP program has been utilized. The accuracy of the computed high resolution airloads (order of 0.5 degs.) required for prediction of impulsive noise is closely examined in terms of the grid clustering in the vicinity of proprotor plane. Fig. 14 depicts a flow chart for the process used in coupling the ROTTILT with the WOPWOP and TIN2 programs.

10 Validations

In order to accurately model the rotor in the ROTTILT-v2 solver, we have conducted a number of computational runs for the isolated XV15 proprotor. Aircraft components such as wing, fuselage, and nacelle were removed from the computational domain before performing additional rotor grid refinements. We have computed the figure-of-merit for the XV15 isolated rotor for a range of Ct/σ from 0.06 to 0.17 which covers the full spectrum of the measured data presented in the reference [10]. Figure (15) depicts the correlation with the measured data which is in very good agreement over a wide range of flight conditions. As shown, an important feature associated with the computed results is the capturing of the rotor stall characteristics observed in the experimental data. The study made here has a significant value in establishing the computational accuracy of the solver before the inclusion of more aircraft components where an accurate assessment of the computed flow could be much more cumbersome.

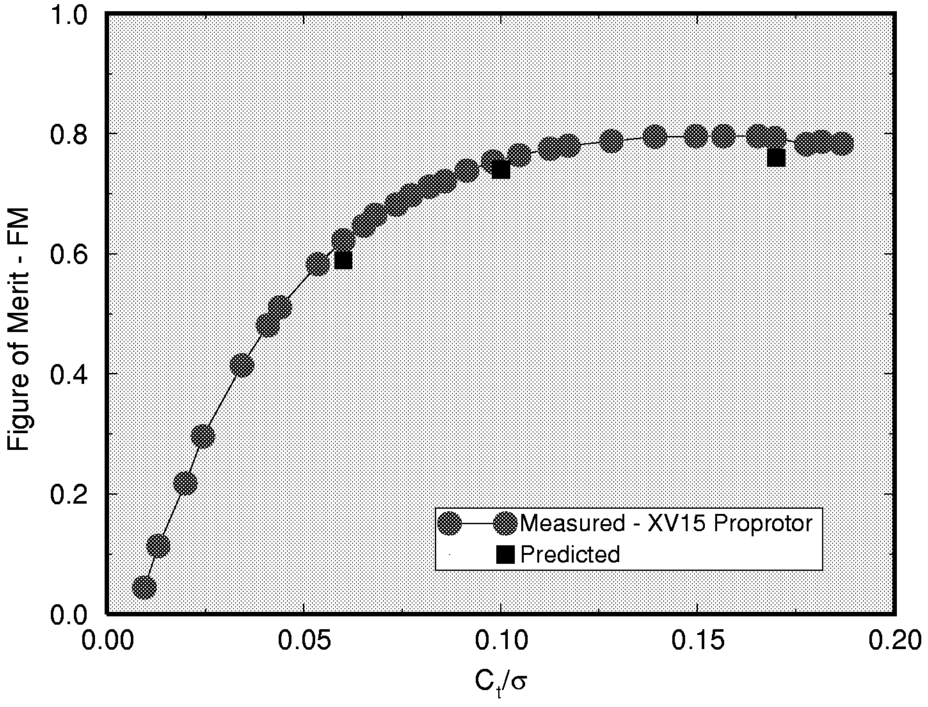

Figure 15: XV15 Isolated Hover Performance

To better understand and evaluate the predictions, the XV15 full-span surface geometry and the flow field are graphically represented. Because of the lack of detailed flow measurement for XV15 aircraft, no attempt is made to justify the accuracy of the computed flow features captured in our CFD simulation of the fountain flow effects. However, it is encouraging to observe that the fountain flow feature associated with the tiltrotor is physical.

Figs. (16 and 17) show the flow field computed by ROTTILT in a constant x-plane which intersects the nacelle and the wing laterally. As shown, the proprotor wake flow features are analyzed using velocity traces (Fig. 16) and velocity vector field (Fig. 17). The fountain flow region on the rotor retreating side (above the wing) has been captured and its extent is clearly defined. Figure 16 shows three distinct regions which can listed as: 1) fountain flow region on the rotor retreating side over the wing upper surface area; 2) typical wake contraction region on the rotor advancing side (clean side); and 3) a stagnated flow region below the wing near the wing fuselage juncture. Furthermore, the rotor downwash flow over the wing region has been split into two distinguishable domains with most of the flow from the wing midspan to the fuselage centerline entrained into the fountain flow region. Slight asymmetry associated with the recirculation region of the fountain between the two rotors is attributed to disparity of the rotor trim forces computed for each proprotor by the ROTTILT solver. A difference of 4 lbs in rotor thrust is obtained from the flow solution for the XV15 proprotor trim analysis after 1200 iterations.

In order to investigate the effects of the fountain flow on the rotor performance, we have presented the results for the rotor inflow velocity variation in terms of rotor azimuth, as shown in Fig (18). In the comparison, the rotor mean inflow velocities for the XV15 isolated rotor case where the aircraft components, fuselage, wing, and nacelle have been removed from the computation domain are compared with the corresponding results from the semi-span XV15 aircraft. Two distinct trends for the inflow velocity are noticeable for the cases considered here where a significant deviation from the mean values is computed in the fountain

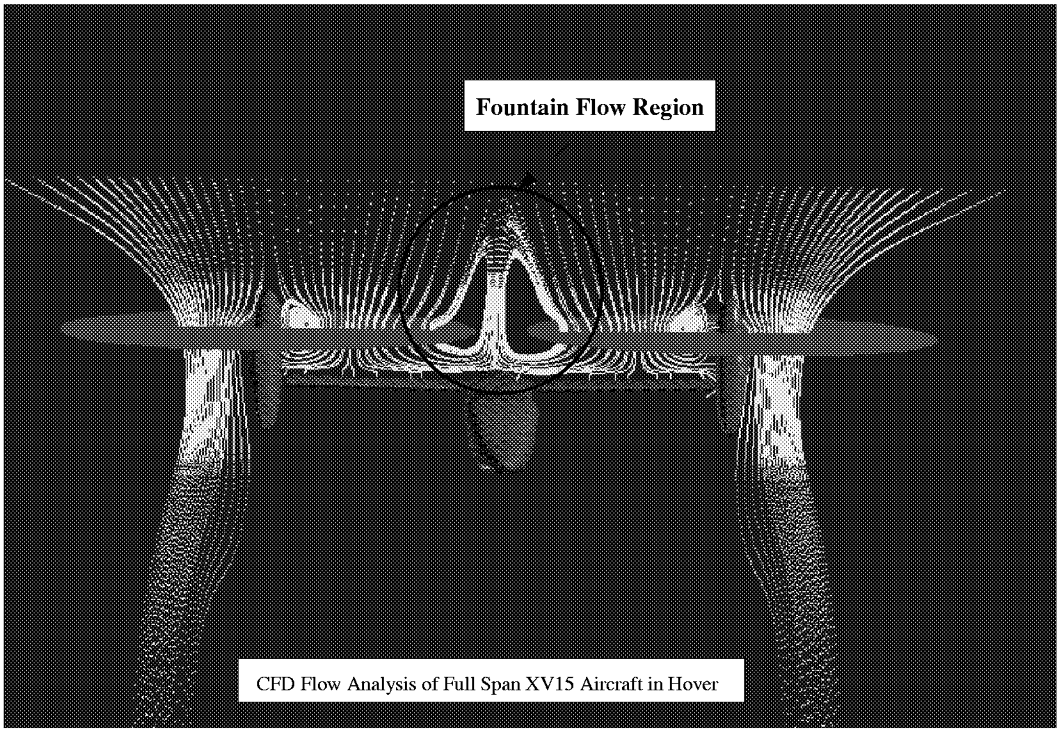

Figure 16: Numerical Simulation of Full Span XV15 Configuration in Hover - ROTTILT Velocity Trace

flow region (between ψ= 230 degs. and ψ= 300 degs.). Consequently, the computed rotor lift as a function of radial stations is affected, as shown in Fig (19). Most importantly, from aerodynamic and acoustic perspectives, the blade airloads have been altered near the tip region for the rotor blade, for example, at ψ= 100 degs. versus ψ= 280 degs.

Bell's hover/inflow turbulence data is employed for ROTTILT inflow validation. Their inflow measurement was made at 7 ins above the rotor on the advancing side (no fountain flow) and the retreating side (above the wing region). Bell's model rotor is a semi-span tiltrotor 15 percent scale of the V22 aircraft with JVX proprotor configuration. In order to correlate the ROTTILT mean inflow computation with that of the measured data the solver with the JVX rotor and semi-span aircraft components was run for up to 1200 flow iterations. The data at the inflow plane corresponding to 7 ins above the rotor plane were extracted from the flow computed by ROTTILT. Figure (20) depicts the comparison of the computed inflow with the measured data. Only the w-component (axial) is presented here since it is considered to be more important parameter than the other two inflow components (namely u and v) to the accuracy of the noise prediction. As shown, most of the salient features of the measured rotor inflow have been captured in the prediction. For example, in the fountain flow region, the computations show a sharp radial variation of the inflow near the tip region changing from a negative value (upwash) outboard of the tip to a large positive (downwash) value which remains virtually constant over most of of the blade span, similar to features observed in the experimental data, see Figure (1). However, the intensity of the dowwash regions ahead and aft of the fountain regions are somewhat underpredicted. In addition, the radial extent of the inflow gradient is limited to a smaller region near the blade tip than that observed in the measured data. One plausible explanation is the lack of accurate modeling of the flow on the plane of symmetry in the absence of bounadry turbulence boundary layer model in the ROTTILT solver. Overall, the correlation with the experimental data is considered to be encouraging with most of the important flow features over rotor plane captured in the flow field computational results.

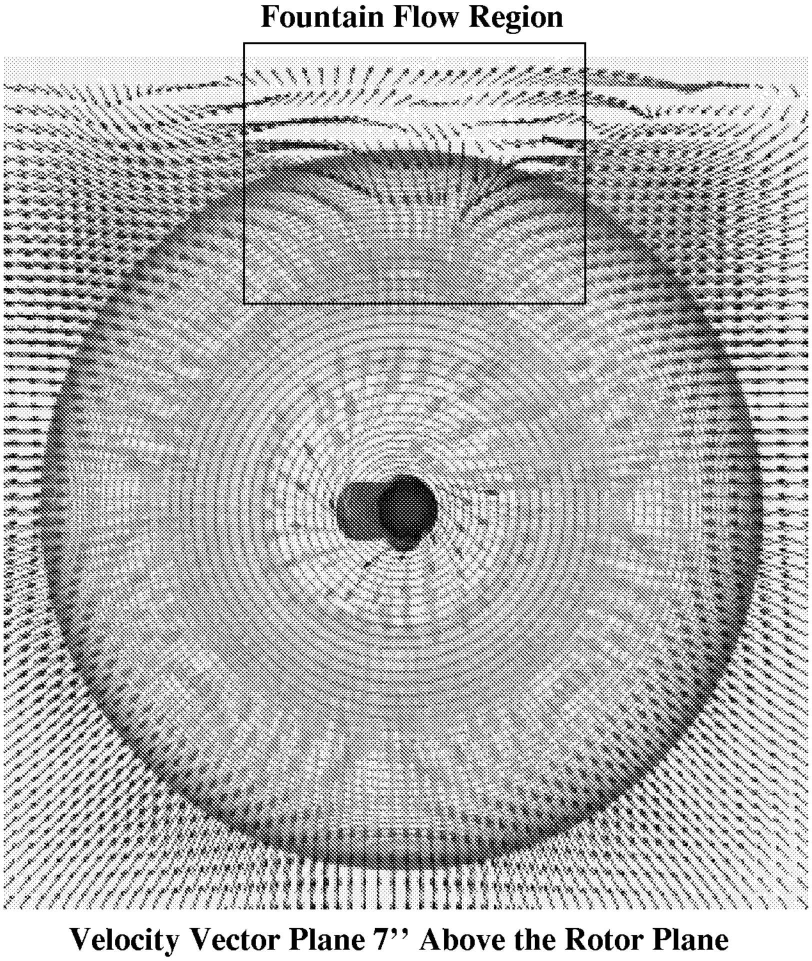

Figure 17: Numerical Simulation of Full Span XV15 Configuration in Hover - ROTTILT Velocity Vectors

Acoustic validation was performed using the flight test data for the XV15 aircraft which presented in Conner's AHS-RaeS Ref[12]. The correlation is made with acoustic time histories only. The XV15 acoustic characteristics are examined by considering the noise radiation pattern at two observer locations corresponding to ϕ = 45 degs. and ϕ=135 degs., see Figure (21). The noise characteristics of titlrotor aircraft can be assessed accurately by these two observers. Moreover, from the flight test data, it is clear that the longitudinal variations of the acoustic characteristics associated with the XV15 aircraft are more dominant than their lateral variations.

Figure(21) depicts measured and predicted acoustic time histories corresponding to the two observer locations. Acoustic analysis was performed using WOPWOP program. The predicted results (microphone: ϕ=45 degs.) contain negative peak of -20 dynes-per-sqcm amplitude which are not clearly detectable in the experimental data. These peaks are associated with the the impulsive loading noise and not the thickness noise component due to the observer location. In the Figure (21), also depicted are the acoustic pressure results for the microphone located at ϕ=135 degs. Overall the comparison with the measured data for this microphone is considered to be encouraging. As shown, the general pulse width has been predicted correctly, whereas the peak-to-peak amplitude is under predicted by 50 percent. In summary, the CFD model ROTTILT coupled with the linearized acoustic model WOPWOP have been effective for aero/acoustic analysis of the tiltrotor in hover. The additional noise source associated with the fountain flow effects which

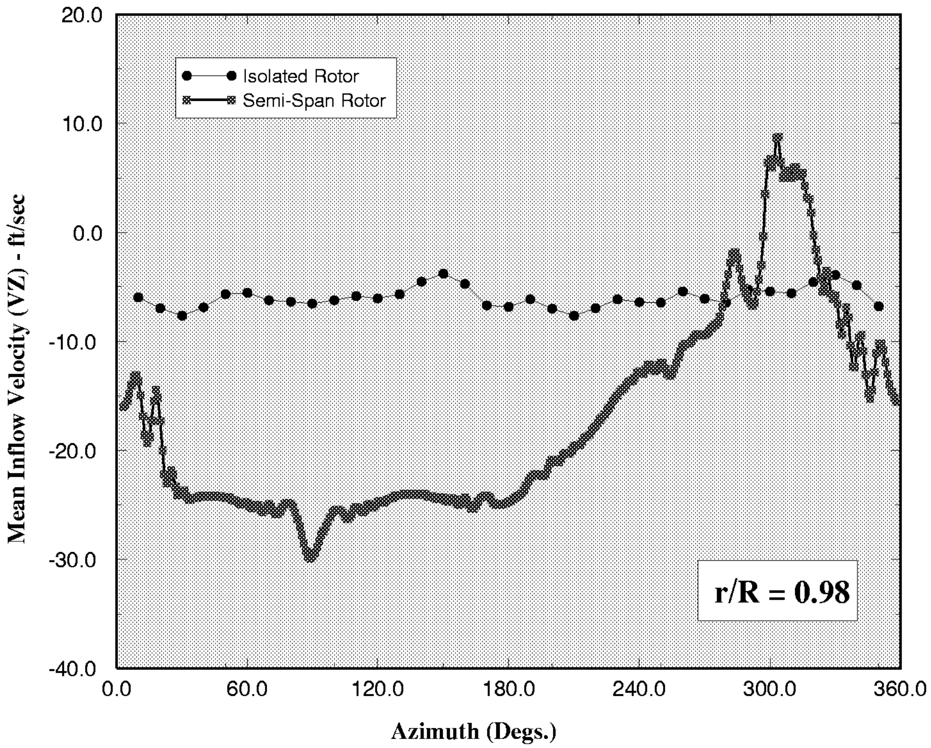

Figure 18: Azimuthal Variation of Rotor Mean Inflow for Full Span XV15 in Hover

is due to turbulence ingestion is not included in the computed results presented in this report.

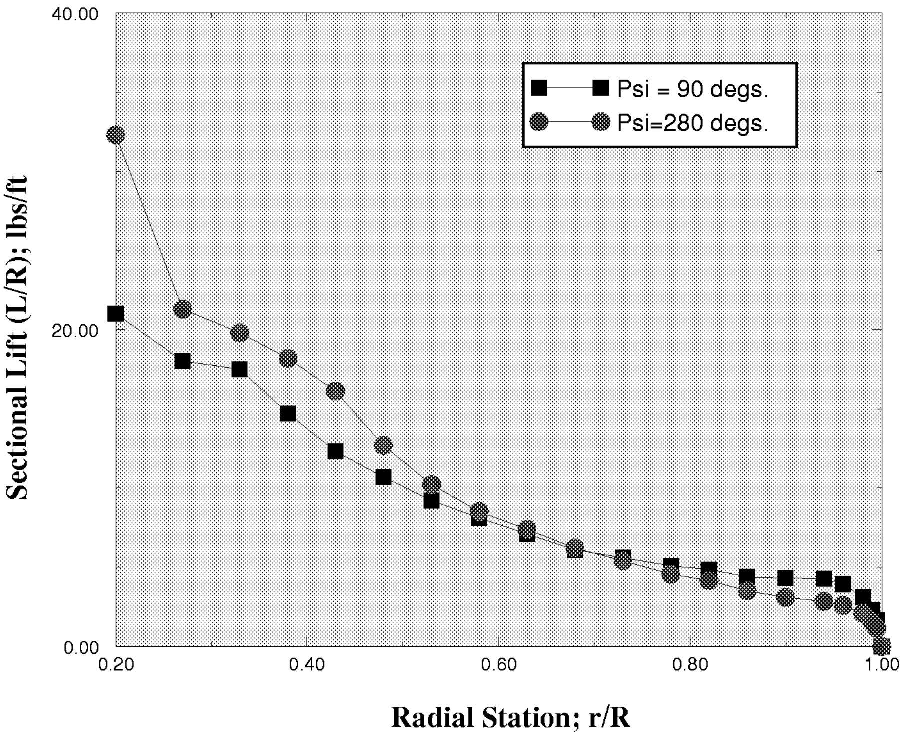

Figure 19: Radial Distribution of Local Inflow Velocity for Full Span XV15 in Hover - ROTTILT Solution

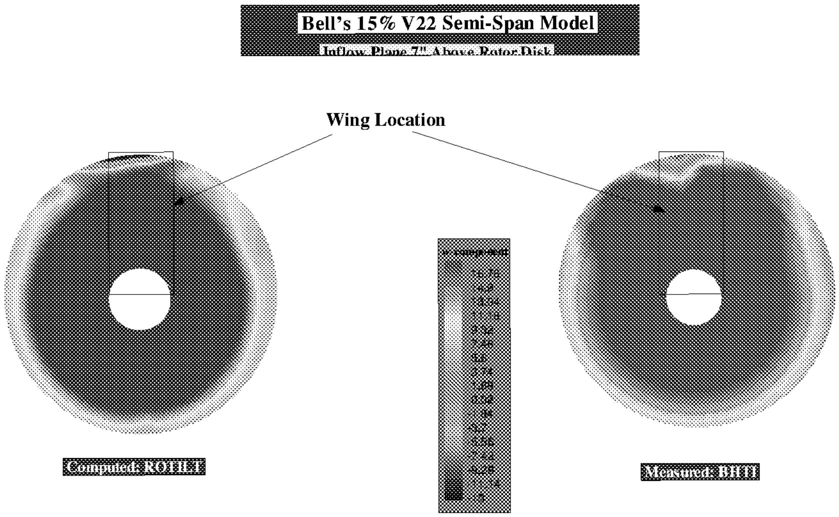

Figure 20: Comparison of Rotor Mean Inflow Velocity for a 15 percent Scaled JVX Rotor

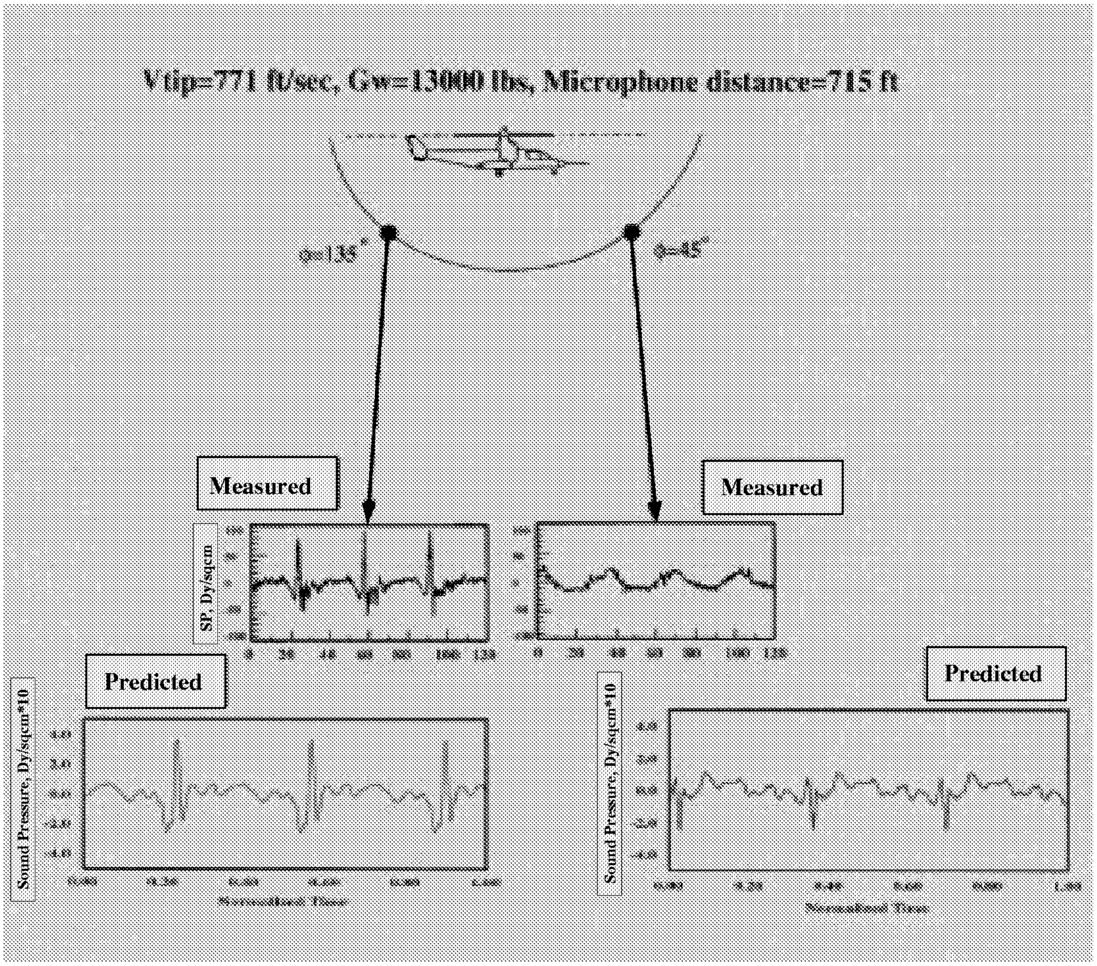

Figure 21: Hover Acoustic Pressures for XV15 Aircraft: Along its Longitudinal Axis

11 References

1. Coffen, C. D., George, A. R., Hardings, H., and Stevenson, R., "Flow Visualization and Flow Field Measurements of a 1/2 Scale Tilt Rotor Aircraft in Hover", Proceedings of the American Helicopter Society Technical Specialist Meeting, October, 1991

2. Rajagoplan R. G. and Mathur S. R. ,"Three dimensional Analysis of a Rotor in Forward Flight", 47th Annual Forum of the AHS, Phoenix, Arizona, May 1991.

3. Brentner, K. S.,"Prediction of Helicopter Rotor Discrete Frequency Noise, - A computer Program Incorporating Realistic Blade Motions and Advanced Acoustic Formulation", NASA-TM-87721, October 1986.

4. Patankar S. V. "Numerical Heat Transfer and Fluid Flow", Hemisphere Publishing Corporation, 1980.

5. Rajagopalan, R. G., and Fanucci, J. B., " Finite Difference Model for Vertical Axis Wind Tunnel Rotors", Journal of Propulsion and Power, Vol. 1, No. 6, November-December 1985.

6. Rajagopalan, R. G., Lim, C. K., " Laminar Flow Analysis of a Rotor in Hover", Journal of the American Helicopter Society, January, 1991.

7. Mathur, S. " Three Dimensional Analysis of a Rotor in Forward Flight", M. S. Thesis, Iowa State University, 1989

8. Hassan, A. A.., Charles, B. D., Tadghighi, H., and Burley, C., "A Consistent Approach for Modeling the Aerodynamics of Self-Generated Rotor Blade-Vortex Interactions", Paper presented at the 49th Annual Forum of the American Helicopter Society, May 1992.

9. George, A. R., Smith, C. A., Maisel, M. D., and Brieger, J. T., "Tiltrotor Aircraft Aeroacoustic", Paper presented at the 45th Annual Forum of the AHS, May 22-24, 1989.

10. Felker, F. F., Young, L. A. and Signor, D. B., "Performance and Loads Data from a Hover Test of a Full-Scale Advanced Technology XV15 Rotor," NASA Technical Memorandum 86854, January 1986.

11. Conner, D. A., Wellman, B., "Far-Field Hover Acoustic Characteristics of the XV-15 Tiltrotor Aircraft with Advanced Technology Blades", AHS/RAeS Technical Specialists Meeting on Rotorcraft Acoustics and Fluid Dynamics, Philadelphia, PA, October 15-16, 1991.

12 Appendices

A Sample Input File

An example input file is provided below, followed by the explanation for each input field-line.

Note: The comment lines are required in the input file.

hxv15.dat

```
_DEBUG_ITEROP_ITERCL_LUNWR_LUNRS_LUNFL_LUNPL_LUNTB_LUNCT
 FALSE ,    20   ,  180   ,    6     ,   8   ,    9    ,   10   ,    17   ,  18

_DETAIL____U_________V_________P_______W_____str Fun____GAM______RO__
 TRUE , TRUE , TRUE , TRUE , TRUE , FALSE , FALSE , FALSE

__LROCON__LMUCON__LSOLID__LDATPR__LUGRID__
   TRUE   ,   TRUE    ,  TRUE ,  FALSE ,  TRUE

__LGEOMP__LUCOFP__LVCOFP__LWCOFP__LPCOFP__LTCOFP__LPLOTP__LFILDP__LPLT3D
FALSE   ,   FALSE ,  FALSE ,  FALSE    ,   FALSE   , FALSE ,   TRUE   ,   TRUE ,  TRUE

__MAX_ITERATIONS___ITEMOD____NU____NV____NW____NP____NT__
            200            ,     10      ,  2  ,    2  ,  2  ,  6  ,  75

__RHO_(0.002377)__MU(3.719E-07)___RELAX-X__RELAX-Y__RELAX-Z__
     0.0023700     ,    3.719E-07     ,  0.05D0   ,  0.05D0  ,   0.05D0

__UINF_____VINF_____WINF____PINF______TINF___REAL-GAS-CON__GAMMA__
 0.0d0   ,  0.0D0 ,  -1.0D0 ,  2116   ,  419.0  ,      1718.0D0       ,  1.4D0

__LINI____LOUTI___LINJ___LOUTJ___LINK__LOUTK___LIP______LJP_____LKP__
 TRUE , TRUE , TRUE , TRUE , TRUE , FALSE , TRUE , TRUE , TRUE

__IPREF__JPREF__KPREF____WINI___WOUTI___WINJ___WOUTJ__WINK__WOUTK
     2    ,    64   ,    59    , FALSE , FALSE , FALSE , FALSE , FALSE , FALSE

__DRELXU__DRELXV__DRELXW__DRELXT__DRELXP__MODREL_RELMAX
     0.01     ,    0.01   ,    0.01    ,    0.01      ,     0.00      ,     50     ,     0.2

__LC81___LSRSEC____LSMSEC___LCLCD____LCLCOR__LROTBOD__
 TRUE , FALSE    ,   FALSE    , TRUE    ,   TRUE    ,     FALSE

__Velocity check__IUCK____JUCK____KWCK___XUCK___YVCK__ZWCK
                         26   ,    22   ,    17   ,      0.0   ,    0.0   ,  0.0

__Number_of_Rotors_____DATA SET REFERENCE =14 NASA(TM X-952)__IBOUND
1 , 4

__Name of airfoil used__

MODIFIED NACA_0012

********************* Data for rotor # 1 ************************************

__Airfoil____# blades____Press dist(r/R)__

     1    ,    3.0D0   ,    0.250D0

Tip speed__Rotor radius_Hub radius(r/R)_Hinge Offse(r/R)_clock rot__
```

```
723.12D0   ,     12.5     ,      0.1353      ,       0.0357       ,  FALSE

Number Of Reference Rotor radius(r/R)max has been set to 45f_______

      44

Reference Rotor radius(r/R)_________________________________________

.100, .125, .150, .175,
.200, .225, .240, .250, .260, .275, .290, .300, .310, .325,
.340, .350, .375, .400, .425, .475, .500, .525, .575, .600,
.625, .650, .675, .700, .750, .775, .800, .825, .850, .875,
.900, .915, .925, .940, .950, .960, .970, .980, .990, .995


CL cor(r/R)___Ref Twist__# Blade Div___Azi cor__# azi loc_____conv fac___
     0.9D0   ,     -0.0D0   ,      100      ,  0.0D0   ,     36      , 0.083333333

# harmonic pit__# harmonic flap__FLAPPING__flap geo corr
          2        ,             3          ,     FALSE   ,   FALSE

–Harmonic_pitching_coefficients–Positive series–
0,       13.35D0,       0.000D0
1,       -0.5D0,       0.300D0
–Harmonic_flapping_coefficients–Positive series–
0       0.0000       0.0000
1       0.0000       0.0000
2       0.0000       0.0000
# data_points____________________________________________________________
21
nn____r/R____deflec____Chord/RAD___CL des___T/chrd___Twist___

 1,   .1353,   0.0D0,       .13780,       0.0E0,   0.12E0,   29.500

 2,   .2035,   0.0D0,       .13780,       0.0E0,   0.12E0,   25.500

 3,   .2700,   0.0D0,       .13780,       0.0E0,   0.12E0,   20.750

 4,   .3250,   0.0D0,       .13470,       0.0E0,   0.12E0,   18.625

 5,   .3750,   0.0D0,       .13470,       0.0E0,   0.12E0,   15.875

 6,   .4250,   0.0D0,       .13470,       0.0E0,   0.12E0,   13.500

 7,   .4750,   0.0D0,       .13470,       0.0E0,   0.12E0,   11.250

 8,   .5250,   0.0D0,       .13470,       0.0E0,   0.12E0,    9.000

 9,   .5750,   0.0D0,       .13470,       0.0E0,   0.12E0,    6.875

10,   .6250,   0.0D0,       .13470,       0.0E0,   0.12E0,    4.875

11,   .6750,   0.0D0,       .13370,       0.0E0,   0.12E0,    2.813

12,   .7250,   0.0D0,       .12330,       0.0E0,   0.12E0,    0.944

13,   .7750,   0.0D0,       .10990,       0.0E0,   0.12E0,   -0.833

14,   .8200,   0.0D0,       .10220,       0.0E0,   0.12E0,   -2.354

15,   .8600,   0.0D0,       .09710,       0.0E0,   0.12E0,   -3.771

16,   .9000,   0.0D0,       .09080,       0.0E0,   0.12E0,   -5.216
```

```
17,  .9350,  0.0D0,     .08330,    0.0E0,  0.12E0,  -6.500

18,  .9600,  0.0D0,     .07170,    0.0E0,  0.12E0,  -7.438

19,  .9750,  0.0D0,     .05730,    0.0E0,  0.12E0,  -8.000

20,  .9850,  0.0D0,     .04770,    0.0E0,  0.12E0,  -8.350

21,  .9950,  0.0D0,     .03810,    0.0E0,  0.12E0,  -8.700

__no of rad stations for airfoil table_____ DATA IS DIFF for non c81__
5

__station at which airfoil table is considered___
0.00, 0.24, 0.92, 0.95, 1.00

__logical unit for the airfoil tables___
28

___The name of the file corresponding to lun = 28___
64-x08.c81

___The name of the file corresponding to lun = 28___
64-x08.c81

___The name of the file corresponding to lun = 28___
64-x12.c81

___The name of the file corresponding to lun = 28___
64-x18.c81

___The name of the file corresponding to lun = 28___
64-x25.c81

LWING__LNACEL__LV22BD
TRUE, TRUE, TRUE

LTRIM__idtriml__ctreq__DELCOLM__TRIMTOL
TRUE , 10 , 0.013d0 , 2d0 , 0.00001
```

In the subsequent pages, the meaning of the field in each line, the type of the variable, and the possible values are explained.

```
_DEBUG_ITEROP_ITERCL_LUNWR_LUNRS_LUNFL_LUNPL_LUNTB_LUNCT
  FALSE ,    20   ,  180   ,    6     ,   8   ,    9    ,   10    ,    17   ,   18
```

This line has 9 fields and the values need to be seperated by comma. The user is advised not to change these values in this line except the value for field DEBUG. Each field is explained below.

Table 4: A1: Input/Output Print Control Parameters

Parameter Name	Values Type	Description
DEBUG	True/False	Whether Debug information is required
ITEROP	Integer	Not Applicable
ITERCL	Integer	Not Applicable
LUNWR	Integer	Logical Unit for standard output
LUNRS	Integer	Logical Unit for Restart files
LUNFL	Integer	Logical Unit for field information
LUNPL	Integer	Logical Unit for plot files
LUNTB	Integer	Logical Unit for files
LUNCT	Integer	Logical Unit for files

```
_DETAIL____U_________V_________P_______W_____str Fun____GAM______RO__
  TRUE , TRUE , TRUE , TRUE , TRUE , FALSE , FALSE , FALSE
```

This line has 8 fields and the values need to be seperated by comma. The fields in this line, allows the user to selectively print the fields, to assist in debugging. This is generally set to FALSE otherwise the output would be very HUGE.

Table 5: A2: Flow Field Print Control Flags

Parameter Name	Values Type	Description
DETAIL	TRUE/FALSE	If FALSE none of the following values will be printed
U	TRUE/FALSE	If TRUE the values of U velocity will be printed
V	TRUE/FALSE	If TRUE the values of V velocity will be printed
P	TRUE/FALSE	If TRUE the values of P Pressure will be printed
W	TRUE/FALSE	If TRUE the values of W velocity will be printed
str Fun	TRUE/FALSE	Not Applicable - Do not Change
GAM	TRUE/FALSE	If TRUE the values of Viscosity will be printed
RO	TRUE/FALSE	If TRUE the values of Density will be printed

```
__LROCON__LMUCON__LSOLID__LDATPR__LUGRID__
   TRUE   ,  TRUE   ,  TRUE ,  FALSE ,  TRUE
```

Table 6: A3: Solid Body Boundary Condition Control Flags

Parameter Name	Values Type	Description
LROCON	TRUE/FALSE	If TRUE Density is a constant
LMUCON	TRUE/FALSE	If TRUE Viscosity is a constant
LSOLID	TRUE/FALSE	If FALSE there is no body For example, in the case of an isolated rotor, LSOLID should be FALSE and for the tilt rotor with a wing, the LSOLID is TRUE as there is a solid body in the field
LDATPR	TRUE/FALSE	When this is TRUE, then the system tries to use the restart file for processing
LUGRID	TRUE/FALSE	Whether User defined special Grid information has been provided to specify the geometry

Note :

- Make sure to set LROCON and LMUCON to TRUE as this version provides for cosntant density and cosntant viscosity.
- LUGRID has a specialised use; Make sure to set LUGRID to TRUE.

```
__LGEOMP__LUCOFP__LVCOFP__LWCOFP__LPCOFP__LTCOFP__LPLOTP__LFILDP__LPLT3D
FALSE   ,  FALSE ,  FALSE ,  FALSE   ,  FALSE   , FALSE ,  TRUE  ,  TRUE ,  TRUE
```

Table 7: A4: Geometry and Flow Print Control Flags

Parameter Name	Values Type	Description
LGEOMP	TRUE/FALSE	Debug Prints Geometry related parameters
LUCOFP	TRUE/FALSE	Debug Prints U Coefficient
LVCOFP	TRUE/FALSE	Debug Prints V Coefficient
LWCOFP	TRUE/FALSE	Debug Prints W Coefficient
LPCOFP	TRUE/FALSE	Debug Prints P Coefficient
LTCOFP	TRUE/FALSE	Debug Prints T Coefficient
LPLOTP	TRUE/FALSE	Debug plotting grid information ASCII output of (X,Y,Z) of the grid
LFILDP	TRUE/FALSE	Debug plotting Field information ASCII output of (U,V,W,P) in the domain
LPLT3D	TRUE/FALSE	If TRUE prints output in PLOT3D format for usage with FAST in two files in binary format The fort.23 and fort.24 files are the grid and the Q-file respectively.

```
__MAX_ITERATIONS___ITEMOD____NU____NV____NW____NP____NT__
          200          ,    10     ,  2  ,   2  ,  2  ,  6  ,  75
```

The meanings of the above fields are explained in the following table.

Table 8: A5: Flow Numerical Iteration Control Variables

Parameter Name	Values Type	Description
MAX_ITERATIONS	Integer	Maximum Number of Iterations
ITEMOD	Integer	Iteration Modulus - After every ITEMOD iterations, restart information will be written to restart file
NU	Integer	Number of U Velocity sweeps for each solver iteration (2 is found to be optimal)
NV	Integer	Number of V Velocity sweeps for each solver iteration (2 is found to be optimal)
NW	Integer	Number of W Velocity sweeps for each solver iteration (2 is found to be optimal)
NP	Integer	Number of P Pressure sweeps for each solver iteration (6 is found to be optimal)
NT	Integer	Do not Change It is used to determine the number of iterations required to stabilize the flowfield

```
__RHO_(0.002377)__MU(3.719E-07)___RELAX-X__RELAX-Y__RELAX-Z__
    0.0023700     ,     3.719E-07     ,   0.05D0   ,   0.05D0  ,    0.05D0
```

In this line, the fluid density ρ and viscosity μ, and the numerical relaxation parameters (for each of the momentum equation in all the three coordinate directions) are entered. Lower Relaxation will result in better stability while higher relaxation may converge faster; suggested initial value for relaxations is 0.05D0. The User will have the flexibility to increase the initial relaxation values (entered in the above line) periodically after a User's specified number of iterations, and also limit the maximum values of these parameters by entering the appropriate fields in subsequent lines.

Table 9: A6: Numerical Relaxation Control Variables

Parameter Name	Values Type	Description
RHO	Real	Density
MU	Real	Kinematic Viscosity
RELAX-X	Real	Relaxation in X diretion
RELAX-Y	Real	Relaxation in Y direction
RELAX-Z	Real	Relaxation in Z direction

```
__UINF_____VINF_____WINF____PINF______TINF___REAL-GAS-CON__GAMMA__
  0.0d0   ,   0.0D0 ,   -1.0D0 ,   2116   ,   419.0  ,       1718.0D0       ,   1.4D0
```

In the above line, the freestream values for the velocity components, pressure, and temperature are entered. For hover calculations, UINF and VINF are set to zero and WINF is set to -1 ft/s at the top boundary. This program is designed for a hovering rotor. The computational domain is shown in the following figure with the global (X,Y,Z) coordinate system. The boundaries of the domain are at distances considered to be numerically infinite. For a hovering rotor, a far upstream velocity of -1 ft/sec is necessary (at Z-max).

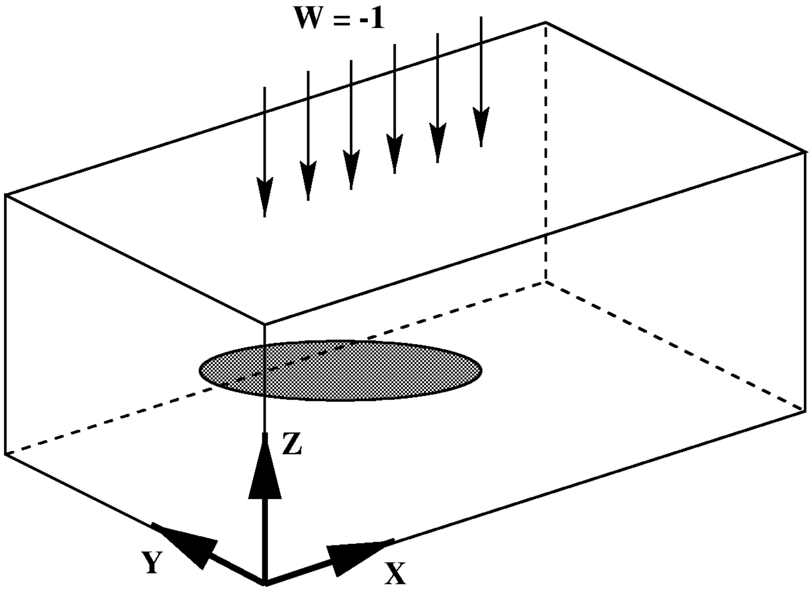

Figure 22: Global coordinates and boundary conditions.

Table 10: A7: Flowfield Domain Boundary Condition Variables

Parameter Name	Values Type	Description
UINF	Real	U Velocity at Infinity
VINF	Real	V Velocity at Infinity
WINF	Real	W Velocity at Infinity
PINF	Real	Pressure at Infinity
REAL-GAS-CON	Real	Gas Constant (R)
GAMMA	Real	Ratio of specific heats

```
__LINI___LOUTI___LINJ___LOUTJ___LINK__LOUTK___LIP______LJP_____LKP__
  TRUE , TRUE , TRUE , TRUE , TRUE , FALSE , TRUE , TRUE , TRUE
```

The first six logical flags in the above allow the User to treat the respective boundaries as symmetric/outflow or specified boundaries. The 'IN' (preceeding I, J, or K) indicates the boundary plane at X, Y, or Z values smaller than the one which are preceeded by the 'OUT'. There must be at least one specified boundary and one outflow boundary.

For the hover case, the top boundary is specified with a velocity of -1 ft/s. Hence, 'LOUTK' is FALSE. The rest of the boundaries are allowed to float and hence all the corresponding logical parameter values are set to TRUE. At a boundary where the velocity is specified and is not allowed to change, this variable should be set to FALSE.

The fields LIP, LJP, and LKP are logical variables which allow the user to sellectively choose directions in which the flow solver has to be inverted. For example, if LIP is set to FALSE, then the inversion routine solves only in the j- and k-directions and not in the i-direction. It is recommended that all values are set to TRUE for the hovering rotor.

Table 11: A8: Flowfield Numerical Solution Directive Control Flags

Parameter Name	Values Type	Description
LINI	TRUE/FALSE	Set to TRUE if X-min is a symmetry or outflow Boundary
LOUTI	TRUE/FALSE	Set to TRUE if X-max is a symmetry or outflow Boundary
LINJ	TRUE/FALSE	Set to TRUE if Y-min is a symmetry or outflow Boundary
LOUTJ	TRUE/FALSE	Set to TRUE if Y-max is a symmetry or outflow Boundary
LINK	TRUE/FALSE	Set to TRUE if Z-min is a symmetry or outflow Boundary
LOUTK	TRUE/FALSE	Set to TRUE if Z-max is a symmetry or outflow Boundary
LIP	TRUE/FALSE	Set to TRUE to solve in I direction
LJP	TRUE/FALSE	Set to TRUE to solve in J direction
LKP	TRUE/FALSE	Set to TRUE to solve in K direction

```
__IPREF__JPREF__KPREF____WINI___WOUTI___WINJ___WOUTJ__WINK__WOUTK
    2   ,   64  ,   59   , FALSE , FALSE , FALSE , FALSE , FALSE , FALSE
```

The meanings of the above fields are adequately explained in the following table.

Table 12: A9: Computational Domain BC Setting Flags

Parameter Name	Values Type	Description
IPREF	Integer	All pressure calculations are relative to the pressure
JPREF	Integer	at a point. The coordinate indexes of this point are
KPREF	Integer	identified by (IPREF,JPREF,KPREF)
WINI	TRUE/FALSE	Set to TRUE if the Xmin-plane is a wall
WOUTI	TRUE/FALSE	Set to TRUE if the Xmax-plane is a wall
WINJ	TRUE/FALSE	Set to TRUE if the Ymin-plane is a wall
WOUTJ	TRUE/FALSE	Set to TRUE if the Ymax-plane is a wall
WINK	TRUE/FALSE	Set to TRUE if the Zmin-plane is a wall
WOUTK	TRUE/FALSE	Set to TRUE if the Zmax-plane is a wall

```
__DRELXU__DRELXV__DRELXW__DRELXT__DRELXP__MODREL_RELMAX
   0.01    ,  0.01  ,   0.01   ,   0.01   ,   0.00   ,   50   ,   0.2
```

Table 13: A10: Numerical Relaxation Control Variables

Parameter Name	Values Type	Description
DRELXU	Real	Relaxation increment for solving the u-velocity
DRELXV	Real	Relaxation increment for solving the v-velocity
DRELXW	Real	Relaxation increment for solving the w-velocity
DRELXT	Real	Relaxation increment for solving the temperature
DRELXP	Real	Relaxation increment for solving the pressure
MODREL	Real	Number of iterations between two successive changes in the relaxations
RELMAX	Real	Upper limit of all the relaxation values

```
__LC81___LSRSEC____LSMSEC___LCLCD____LCLCOR__LROTBOD__
  TRUE , FALSE    ,   FALSE   , TRUE   ,   TRUE   ,    FALSE
```

Table 14: A11: Rotor Airfoils Configuration Flags

Parameter Name	Values Type	Description
LC81	TRUE/FALSE	Whether the Airfoil is of type c81 If TRUE indicates that the airfoil is of type c81
LSRSEC	TRUE/FALSE	If TRUE indicates that the entire rotor is represented by one airfoil table
LSMSEC	TRUE/FALSE	If TRUE the airfoil tables have only one Mach number entry
LCLCD	TRUE/FALSE	Inactive and do not change
LCLCOR	TRUE/FALSE	Inactive and do not change
LROTBOD	TRUE/FALSE	Inactive and do not change

```
__Velocity check__IUCK____JUCK____KWCK___XUCK___YVCK__ZWCK
                   26   ,   22   ,   17   ,    0.0   ,   0.0   ,  0.0
```

The meanings of the above fields are explained in the following table.

Table 15: A12: Flowfield Solution Directive Monitoring Variables

Parameter Name	Values Type	Description
IUCK	Integer	I Grid Point where we monitor the velocity
JVCK	Integer	J Grid Point where we monitor the velocity
KWCK	Integer	K Grid Point where we monitor the velocity
XUCK	Real	X coordinate where we need to print some value /* NOT USED */
YVCK	Real	Y coordinate where we need to print some value /* NOT USED */
ZWCK	Real	Y coordinate where we need to print some value /* NOT USED */

```
__Number_of_Rotors_____DATA SET REFERENCE =14 NASA(TM X-952)__IBOUND
1 , 4
```

Table 16: A13: Number of Rotor Configuration Control

Parameter Name	Values Type	Description
Number of Rotors	Integer	Number of Rotors
IBOUND	Number	4 is used for representing the boundary conditions for a hovering rotor

```
__Name of Airfoil Used__
MODIFIED NACA_0012
```

Table 17: A14: Rotor Airfoil Identification Command

Parameter Name	Values Type	Description
Name of Airfoil Used	Character	Name of Airfoil Used

********************** Data for rotor # 1 ************************************

Example:

```
__Airfoil____# blades____Press dist(r/R)__
    1     ,     3.0D0    ,     0.250D0
```

Table 18: A15: Rotor Blade Characteristics

Parameter Name	Values Type	Description
Number of Airfoils	Integer	Number of Airfoils
Number of Blades	Real	Number of Blades
r/R	Real	A contant used in tip correction Do not change

```
Tip speed__Rotor radius_Hub radius(r/R)_Hinge Offse(r/R)_clock rot__
723.12D0   ,     12.5     ,      0.1353       ,       0.0357       ,   FALSE
```

Table 19: A16: Rotor Configuration

Parameter Name	Values Type	Description
Tip Speed	Real	Linear velocity of the tip
Rotor Radius	Real	Rotor Radius
Hub Radius	Real	Hub Radius
Hinge Offset	Real	Hinge Offset
clock rot	TRUE/FALSE	Clock rotation If TRUE implies that the rotor is spinning clockwise

```
Number Of Reference Rotor radius(r/R) (max has been set to 45)_______
44
```

Table 20: A17: Rotor Blade Aerodynamic Control Point Setting

Parameter Name	Values Type	Description
Number of Reference Rotor radius	Integer	Number of Reference Rotor radius(r/R)

```
Reference Rotor radius(r/R)________________________________________

.100, .125, .150, .175,
.200, .225, .240, .250, .260, .275, .290, .300, .310, .325,
.340, .350, .375, .400, .425, .475, .500, .525, .575, .600,
.625, .650, .675, .700, .750, .775, .800, .825, .850, .875,
.900, .915, .925, .940, .950, .960, .970, .980, .990, .995
```

```
CL cor(r/R)___Ref Twist__# Blade Div___Azi cor__# azi loc_____conv fac___
    0.9D0   ,    -0.0D0   ,     100     ,  0.0D0   ,    36     , 0.083333333
```

Table 21: A18: Rotor Blade Geometry and Operating Controls

Parameter Name	Values Type	Description
CL cor(r/R)	Real	Tip correction starts at this radius
Ref Twist	Real	Reference Twist
# Blade Div	Number	Number of Blade Divisions
Azi cor	Real	Azimuthal Corrections
# azi loc	Number	Number of azimuth locations at which output is desired
conv fac	Real	conversion factor Inactive and do not change

```
# harmonic pit__# harmonic flap__FLAPPING__flap geo corr
        2       ,          3        ,    FALSE   ,  FALSE
```

Table 22: A19: Rotor Blade Flap and Pitich Motions

Parameter Name	Values Type	Description
# harmonic pit	Number	Number of entries in the harmonic pitch table
# harmonic flap	Number	Number of entries in the harmonic flap table
FLAPPING	TRUE/FALSE	If TRUE the rotor is allowed to flap
flap geo corr	TRUE/FALSE	If TRUE flap Geometry correction is made

The following table contains as many entries as they are stated for the harmonic pitch. In the above example, two entries are specified for the harmonic pitch table. The first value (a_o) refers to the collective pitch. All the values are given in radians and the series is assumed to be positive of the form:

$$\theta = a_o + \sum_{n=1}^{N} a_n cos^n \psi + \sum_{n=1}^{N} b_n sin^n \psi \ . \tag{1}$$

```
–Harmonic_pitching_coefficients–Positive series–
0,      13.35,       0.000
1,      -0.50,       0.300
```

Similarly, three entries are specified for the harmonic flap table in the example above. The first value (c_o) refers to the cone and the other terms belong to the positive series. All values are given in radians and take the form:

$$\beta = c_o + \sum_{n=1}^{N} c_n cos^n \psi + \sum_{n=1}^{N} d_n sin^n \psi \ . \tag{2}$$

```
–Harmonic_flapping_coefficients–Positive series–
0       0.0000       0.0000
1       0.0000       0.0000
2       0.0000       0.0000
```

```
# data_points________________________________________________________
21
```

```
nn____r/R____deflec____Chord/RAD___CL des___T/chrd___Twist___
```

Table 23: A20: Rotor Blade Geometry Definitions

Parameter Name	Values Type	Description
# data_points	Integer	Number of data entries in the table that describes the geometric parametes of the rotor blade
nn	Integer	The table entry
r/R	Real	Non-dimensionalized radius
deflec	Real	Out of plane deflection in degrees
Chord/RAD	Real	Chord non-dimensionalized by radius
CL des	Real	Inactive
T/chrd	Real	Inactive
Twist	Real	Geometric twist of the rotor

```
__no of rad stations for airfoil table_____ DATA IS DIFF for non c81__
5
```

Table 24: A21: Rotor Aifoils Distribution Definitions

Parameter Name	Values Type	Description
no of rad stations for airfoil table	Integer	Number of radial stations for airfoil table

```
__station at which airfoil table is considered___
0.00, 0.24, 0.92, 0.95, 1.00
```

Table 25: A22: Rotor Blade Airfoil Radial Distribution Control

Parameter Name	Values Type	Description
station at which airfoil table is considered	Real	The actuals station at which airfoil table is considered Must match the number specified in the previous line Each value must be separated by a comma

```
__logical unit for the airfoil tables___
28
```

```
___The name of the file corresponding to lun = 28___
64-x08.c81
```

```
___The name of the file corresponding to lun = 28___
64-x08.c81
```

___The name of the file corresponding to lun = 28___
64-x12.c81

___The name of the file corresponding to lun = 28___
64-x18.c81

___The name of the file corresponding to lun = 28___
64-x25.c81

Table 26: A23: Rotor C81 Table File(s) Designation

Parameter Name	Values Type	Description
logical unit for the airfoil tables	Integer	Logical unit for the airfoil tables
Name of the lun = 28 file	Integer	The name of the file containing Airfoil tables; one such line must be provided for each Airfoil table

_LWING__LNACEL__LV22BD
TRUE, TRUE, TRUE

Table 27: A24: Aircraft Components Declaration

Parameter Name	Values Type	Description
LWING	TRUE/FALSE	If TRUE wing is present
LNACEL	TRUE/FALSE	If TRUE nacelle is present
LV22BD	TRUE/FALSE	If TRUE fuselage is present

_LTRIM__idtriml__ctreq__DELCOLM__TRIMTOL
TRUE , 10 , 0.013 , 2.0 , 0.00001

Table 28: A25: Aircraft Trim Conditions

Parameter Name	Values Type	Description
LTRIM	TRUE/FALSE	If TRUE the rotor is trimmed for a given thrust
idtriml	Integer	Number of iterations between trim
ctreq	Real	Non-dimensional thrust required
DELCOLM	Real	Maximum collective change permitted during trim
TRIMTOL	Real	Trim tolerance for θ collective

B ROTTILT Input Geometry Scale Down or Up Program

```
*     -*-FORTRAN-*-*
*     **************************************************************
*     *                                                            *
*     *  ********************************************************  *
*     * *         Developed by Hormoz Tadghighi                * *
*     * *                                                      * *
*     * *             Boeing Helicopter Company                * *
*     * *                 Mesa,  Arizona                       * *
*     * *                                                      * *
*     * *       Output: Scale Geometry from Full Scale         * *
*     * *               to Model Scale                         * *
*     *  ********************************************************  *
*     *                                                            *
*     **************************************************************
*

*-----------------------------------------------------------------------
Program Main
parameter (mi = 300, mj = 300, mk = 300)
dimension xf(mi),yf(mi,mj),zf(mi,mj)
dimension xwu(mi,mj),ywu(mi),zwu(mi,mj)
dimension xwl(mi,mj),ywl(mi),zwl(mi,mj)
dimension xn(mi,mj),yn(mi,mj),zn(mi)
c Input Files
c
open(10, file='fsl.OUT',status='unknown')
open(11, file='wng.OUT',status='unknown')
open(12, file='ncl.OUT',status='unknown')
c Output Files
c
open(14, file='fsl-plot.OUT',status='unknown')
open(15, file='wngupper-plot.OUT',status='unknown')
open(17, file='wnglower-plot.OUT',status='unknown')
open(16, file='ncl-plot.OUT',status='unknown')
c
c Read
c
c Fuselage Sections
c

read (10,*)imax1,jmax1
c
do k = 1, imax1
read(10,*)xf(k)
do j = 1, jmax1
read(10,*)yf(j,k),zf(j,k)
end do
end do
close (10)
c
c
```

```
c Wing Sections
c

        read (11,*)imax2,jmax2
c
do k = 1, imax2
read(11,*)ywu(k)
do j = 1, jmax2
read(11,*)xwu(j,k),zwu(j,k)
end do
end do

        read (11,*)imax4,jmax4
c
do k = 1, imax4
read(11,*)ywl(k)
do j = 1, jmax4
read(11,*)xwl(j,k),zwl(j,k)
delta = zwl(j,K) - zwu(j,k)
        zwl(j,k) = zwu(j,k) + delta
end do
end do
close(11)
c
c
c Nacelle Sections
c

read (12,*)imax3,jmax3
c
do k = 1, imax3
read(12,*)zn(k)
do j = 1, jmax3
read(12,*)xn(j,k),yn(j,k)
end do
end do
close (12)
c
c Write Output Files - Fuselage
c
do k = 1,imax1
do j = 1,jmax1
write(14,1001)xf(k),yf(j,k),zf(j,k)
end do
end do
c
c
c
c Write Output Files - wing
c
do k = 1,imax2
do j = 1,jmax2
write(15,1001)xwu(j,k),ywu(k),zwu(j,k)
end do
end do

do k = 1,imax4
do j = 1,jmax4
write(17,1001)xwl(j,k),ywl(k),zwl(j,k)
end do
end do
c
c
c Write Output Files - nacelle
c
do k = 1,imax3
do j = 1,jmax3
write(16,1001)xn(j,k),yn(j,k),zn(k)
end do
end do
c
 1001 format(2x,f14.5,1x,f14.7,1x,f14.7)
c
close (14)
close (15)
close (16)
stop
end
c-------Input Namelist-----------------------------
 &INPUTS
 NB = 3,
 CMEAN = 1.7225,
 RHO = 1.234,
 RADIUS = 3.81,
 VTIP = 220.41,
 OMEGA = 57.84,
 FC = 30.,
 FCC = 10.,
 &END
```

C ROTTILT Grid Conversion into a 2D Gnuplot's Format

```
*     -*-FORTRAN-*-*
*     **************************************************************
*     *                                                            *
*     *  ******************************************************    *
*     *  *         Developed by Hormoz Tadghighi               *   *
*     *  *                                                     *   *
*     *  *             Boeing Helicopter Company               *   *
*     *  *                 Mesa,  Arizona                      *   *
*     *  *                                                     *   *
*     *  *       Output: Files for Gnuplot in 2D Format        *   *
*     *  *               Grid Generated by ROTTILT Solver      *   *
*     *  ******************************************************    *
*     *                                                            *
*     **************************************************************
*

*-----------------------------------------------------------------------
      Program Main
      integer lu, luu, N, M, N1, N2, N3, M1, M2, M3
      real*8 X(10000,10),Y(10000,10)
      lu = 39
      open (39, file='grid-2d-xyz.dat', status='unknown')
      open (40, file='xygrid.out', status='unknown')
      open (41, file='xzgrid.out', status='unknown')
      open (42, file='yzgrid.out', status='unknown')
      do i = 1 ,3
      Read (lu,15) N, M
      if(i .eq. 1)N1 = N
      if(i .eq. 1)M1 = M
      if(i .eq. 2)N2 = N
      if(i .eq. 2)M2 = M
      if(i .eq. 3)N3 = N
      if(i .eq. 3)M3 = M
      do k = 1, M
         do j = 1, N
            read (lu,20) X(j,i), Y(k,i)
         end do
      end do
      read(lu,101)
      end do
      luu = 39
      do i = 1 ,3
      luu = luu + 1
      if(i .eq. 1)N = N1
      if(i .eq. 1)M = M1
      if(i .eq. 2)N = N2
      if(i .eq. 2)M = M2
      if(i .eq. 3)N = N3
      if(i .eq. 3)M = M3
      write (luu,15) N, M
      do k = 1, M
```

```
        do j = 1, N
           write (luu,20) X(j,i), Y(k,i)
        end do
     end do
     end do

15   format (I25, I25)
20   format(1X,2(1X,F20.10))
101  format(4a20)

     close (39)
     close (40)
     close (41)
     close (42)

     stop
     end
```

D Geometry Section Cuts Generation Program

```
*     -*-FORTRAN-*-*
*     **************************************************************
*     *                                                            *
*     *  ********************************************************  *
*     *  *         Developed by Ganesh & Tadghighi              *  *
*     *  *                                                      *  *
*     *  *             Boeing Helicopter Company                *  *
*     *  *                 Mesa,  Arizona                       *  *
*     *  *                                                      *  *
*     *  *       Output: Conversion of ROTTILT Performance      *  *
*     *  *               Output file for TIN2 & WOPWOP          *  *
*     *  *               Programs & 2D Plots                    *  *
*     *  ********************************************************  *
*     *                                                            *
*     **************************************************************
*

*-----------------------------------------------------------------------
      program Spline

      call Tiltrotor

      stop
      end
c_________________________________________
      subroutine Tiltrotor

      implicit integer(i-n)
      implicit real*4(a-h,o-z)
      character*60 dir,fname,outname

      dimension pt1(200),pt2(200),pt3(200),ptnew1(5000)
      dimension ptnew2(5000),ptnew3(5000),index(200)
      dimension rho(200),tau(200),s(200)

      print *, 'WHAT FILE DO YOU WANT TO MANIPULATE?'
      read(*,'(a30)') fname

      print *, 'HOW MANY CROSS-SECTIONS DO YOU WANT?'
      read *, nxsect

      print *, 'IN WHAT PLANES ARE YOUR CROSS-SECTIONS LOCATED'
      print *, '<xdir>, <ydir>, or <zdir>'
      read(*,'(a4)') dir

      open(1,file=fname,status='old')

c     THE FOLLOWING LINE READS IN THE HEADER THAT SHOULD BE
c     INCLUDED IN THE FILE READ THE NUMBER OF LINES OF DATA
c     (NLINES) AND THE NUMBER OF DATA POINTS PER LINE (MAX).
```

```
      read(1,*) max, nlines


      nx = 0
      nz = 0
      do 50 line = 1,nlines

         if (dir.eq.'xdir') then
            do 10 lninc = 1,max
              read(1,*) pt3(lninc),pt1(lninc),pt2(lninc)
 10         continue
         end if

         if (dir.eq.'ydir') then
            do 11 lninc = 1,max
               read(1,*) pt1(lninc),pt3(lninc),pt2(lninc)
 11         continue
         end if

         if (dir.eq.'zdir') then
            do 12 lninc = 1,max
               read(1,*) pt1(lninc),pt2(lninc),pt3(lninc)
 12         continue
         end if

c        make a pattern of y coordinates from the first set
         if (line.eq.1) then
            step = (pt3(max)-pt3(1))/real(nxsect-1)
            do 20 lninc=1,nxsect
               ptnew3(lninc) = pt3(1)+(lninc-1)*step
 20         continue

         end if

c        create the new x - coordinates along the line
         call spcoef(max,pt3,pt1,s,index,rho,tau,nxsect)
         do 30 lninc = 1,nxsect
            nx = nx+1
            ptnew1(nx) = spline(max,pt3,pt1,s,index,ptnew3(lninc))
 30      continue

c        create the new z - coordinates along the line
         call spcoef(max,pt3,pt2,s,index,rho,tau,nxsect)
         do 40 lninc = 1,nxsect
            nz = nz+1
            ptnew2(nz) = spline(max,pt3,pt2,s,index,ptnew3(lninc))
 40      continue


 50   continue

      close(1)
c     TRANSFORM THE DATA TO READABLE OUTPUT TO THE DATA FILE IN THE
c     FORM OR Y LOCATION OF THE CROSS SECTION AND THE X AND Z
```

```
c     COORDINATES WITHIN THE SECTION.

      print *, 'WHAT IS THE NAME OF THE OUTPUT FILE?'
      read(*,'(a30)') outname
      open(2,file=outname,status='unknown')

      write(2,70) nxsect,nlines
 70   format(1x,2(i3,3x))

      ncount = 0
      do 90 lninc = 1,nxsect
         write(2,100) ptnew3(lninc)
         do 80 line = 1,nlines
            write(2,110) ptnew1(lninc+nxsect*(line-1)),ptnew2(lninc
     +                   +nxsect*(line-1))
 80      continue
 90   continue

 100  format(1x,f10.5)
 110  format(1x,2(f10.5,4x))

      close(2)

      end
c
c_________________________________________
c
       subroutine spcoef(n,xn,fn,s,index,rho,tau,npts)
       implicit real*4 (a-h,o-z)
       dimension xn(n),fn(n),s(n),index(n),rho(npts),tau(npts)
       nm1=n-1
       do 1 i=1,n
   1   index(i)=i
       do 3 i=1,nm1
       ip1=i+1
       do 2 j=ip1,n
       ii=index(i)
       ij=index(j)
       if(xn(ii).le.xn(ij)) go to 2
       itemp=index(i)
       index(i)=index(j)
       index(j)=itemp
   2   continue
   3   continue
       nm2=n-2
       rho(2)=0.0e+00
       tau(2)=0.0e+00
       do 4 i=2,nm1
       iim1=index(i-1)
       ii=index(i)
       iip1=index(i+1)
       him1=xn(ii)-xn(iim1)
       hi=xn(iip1)-xn(ii)
       temp=(him1/hi)*(rho(i)+2.0e+00)+2.0e+00
      rho(i+1)=-1.0e+00/temp
      d=6.0*((fn(iip1)-fn(ii))/hi-(fn(ii)-fn(iim1))/him1)/hi
   4  tau(i+1)=(d - him1*tau(i)/hi)/temp
      s(1)=0.0e+00
      s(n)=0.0e+00
      do 5 i=1,nm2
      ib=n-i
   5  s(ib)=rho(ib+1)*s(ib+1)+tau(ib+1)
      return
      end
c__________________________________________________________
      function spline(n,xn,fn,s,index,x)
      implicit real*4 (a-h,o-z)
      dimension xn(n),fn(n),s(n),index(n)

      i1=index(1)
      if(x.ge.xn(i1)) go to 1
      i2=index(2)
      h1=xn(i2)-xn(i1)
      spline=fn(i1)+(x-xn(i1))*((fn(i2)-fn(i1))/h1-h1*s(2)/6.0)
      return
   1  in=index(n)
      if(x.le.xn(in)) go to 2
      inm1=index(n-1)
      hnm1=xn(in)-xn(inm1)
      spline=fn(in)+(x-xn(in))*((fn(in)-fn(inm1))/hnm1+hnm1*s(n-1)/6.0)
      return
   2  do 3 i=2,n
      ii=index(i)
      if(x.le.xn(ii)) go to 4
   3  continue
   4  l=i-1
      il=index(l)
      ilp1=index(l+1)
      a=xn(ilp1)-x
      b=x-xn(il)
      hl=xn(ilp1)-xn(il)
      spline=a*s(l)*(a**2/hl-hl)/6.0+b*s(l+1)*(b**2/hl-hl)/
     ; 6.0+(a*fn(il)+b*fn(ilp1))/hl
      return
      end
```

E ROTTILT-WOPWOP-TIN2 Hi-Res Airloads and Inflow Velocities Conversions

```
*     -*-FORTRAN-*-*
*     **************************************************************
*     *                                                            *
*     *  ********************************************************  *
*     *  *         Developed by Hormoz Tadghighi                *  *
*     *  *                                                      *  *
*     *  *             Boeing Helicopter Company                *  *
*     *  *                 Mesa,  Arizona                       *  *
*     *  *                                                      *  *
*     *  *       Output: Conversion of ROTTILT Performance      *  *
*     *  *               Output file for TIN2 & WOPWOP          *  *
*     *  *               Programs & 2D Plots                    *  *
*     *  ********************************************************  *
*     *                                                            *
*     **************************************************************
*

*-----------------------------------------------------------------------
program main
        implicit double precision (a-h,o-z)
        logical acou,plot,flow
integer npsi,nrad
dimension rad(361),azim(361),cl(361,361),
     >  cd(361,361),slift(361,361),sdrag(361,361),
     >  sload(361,361),thrust(361,361),torque(361,361),
     >  alpha(361,361),beta(361,361),twist(361,361),
     >  vr(1000,361),vtheta(361,361),vz(361,361),
     >  FZ(361,361),FN(361,361),fact(361),parm(361),
     >  dum(361),t(361),df(361),d(361),VZZ(361),
     >  r1(3),r2(3),theta1(3),theta2(3)
        data acou /.true./
data plot /.false./
data flow /.true./
C***********************************
      NAMELIST/INPUTS/ nb, cmean, rho, radius, vtip, omega, fc, fcc
C***********************************
OPEN(5,FILE='Conv.nam', status='unknown',
     >       FORM='FORMATTED')
OPEN(8,FILE='rot-az-perf.dat', status='unknown',
     >       FORM='FORMATTED')
OPEN(12,FILE='xv15.lds', status='unknown',
     >       FORM='FORMATTED')
OPEN(13,FILE='Vz.inflow', status='unknown',
     >       FORM='FORMATTED')
OPEN(14,FILE='Vz.plot', status='unknown',
     >       FORM='FORMATTED')

pi = 4.*atan(1.)
DTR = 0.01745
```

```
c-------------set rotor sub-domain -------------
c sub domain 1
        r1(1) = 0.1
        r2(1) = 0.3
        theta1(1) = 0.
        theta2(1) = 360.
c sub domain 2
        r1(2) = 0.3
        r2(2) = 0.6
        theta1(2) = 0.
        theta2(2) = 360.
c sub domain 3
        r1(3) = 0.6
        r2(3) = 0.1
        theta1(3) = 0.
        theta2(3) = 360.

c-------------input read
C
C..Read in the input and write it back out
C
        READ (5, INPUTS)
        WRITE (6, INPUTS)
C
read(8,1006)
read(8,1006)
read(8,1006)
        read(8,1001)L, nrad , npsi
        write(6,*)L,nrad , npsi

do 2001 i=1,nrad-1
read(8,1002)
read(8,1003)rad1,rad(i)
read(8,1004)
do 2002 j=2,npsi+1

if(i .lt. 36)
     >  read(8,1005)azim(j),cl(i,j),cd(i,j),slift(i,j),sdrag(i,j),
     >  sload(i,j),thrust(i,j),torque(i,j),alpha(i,j),beta(i,j),
     >  twist(i,j),vr(i,j),vtheta(i,j),vz(i,j)

if(i .ge. 36)
     >  read(8,1007)azim(j),cl(i,j),cd(i,j),slift(i,j),sdrag(i,j),
     >  sload(i,j),thrust(i,j),torque(i,j),alpha(i,j),beta(i,j),
     >  twist(i,j),vr(i,j),vtheta(i,j),vz(i,j),dummy

 2002 continue
 2001 continue
c-----------------------------------------------------------------
c_ perform Smoothing
c
        nd = npsi+1
        azim(1) =0.0
dt = azim(2) - azim(1)
```

```
fc = fc/360.
fcc = fcc/360.

        write(6,*)' Sample interval = ',DT
        DELF=360/(azim(npsi+1)-azim(1))
        write(6,*)' Frequency resolution = ',DELF,' per rev'
        FMAX=DELF*(npsi-1)/2
        write(6,*)' Max. frequency = ',FMAX,' per rev'

do i=1,nrad-1
        cl(i,1) = cl(i,npsi+1)
        cd(i,1) = cd(i,npsi+1)
        slift(i,1) = slift(i,npsi+1)
        sdrag(i,1) = sdrag(i,npsi+1)
        sload(i,1) = sload(i,npsi+1)
        thrust(i,1) = thrust(i,npsi+1)
        torque(i,1) = torque(i,npsi+1)
        alpha(i,1) = alpha(i,npsi+1)
        beta(i,1) = beta(i,npsi+1)
        twist(i,1) = twist(i,npsi+1)
        vr(i,1) = vr(i,npsi+1)
        vtheta(i,1) = vtheta(i,npsi+1)
        vz(i,1) = vz(i,npsi+1)
        enddo

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = cl(i,j)
        enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        cl(i,j) = df(j)
enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do I=1,nd
ch           write(1,*)T(I),DUM(I),DF(I)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = cd(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
```

```
        cd(i,j) = df(j)
        enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do I=1,nd
ch           write(1,*)T(I),DUM(I),DF(I)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = slift(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        slift(i,j) = df(j)
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do Ii=1,nd
ch           write(1,*)T(Ii),DUM(Ii),DF(Ii)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End
        enddo

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = sdrag(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        sdrag(i,j) = df(j)
        enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do II=1,nd
ch           write(1,*)T(II),DUM(II),DF(II)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End
```

```
c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = sload(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        sload(i,j) = df(j)
        enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do I=1,nd
ch           write(1,*)T(I),DUM(I),DF(I)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = thrust(i,j)
enddo
        call smooth (nd,dt,fcc,dum,df)
do j = 1, npsi+1
        thrust(i,j) = df(j)
        enddo
        enddo

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = torque(i,j)
enddo
        call smooth (nd,dt,fcc,dum,df)
do j = 1, npsi+1
        torque(i,j) = df(j)
        enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do I=1,nd
ch           write(1,*)T(I),DUM(I),DF(I)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End
```

```
c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = alpha(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        alpha(i,j) = df(j)
        enddo
        enddo

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = beta(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        beta(i,j) = df(j)
        enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do I=1,nd
ch           write(1,*)T(I),DUM(I),DF(I)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = twist(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        twist(i,j) = df(j)
        enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do I=1,nd
ch           write(1,*)T(I),DUM(I),DF(I)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End
```

```
c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = vr(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        vr(i,j) = dum(j)
        enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do I=1,nd
ch           write(1,*)T(I),DUM(I),DF(I)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = vtheta(i,j)
enddo
        call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
        vtheta(i,j) = df(j)
        enddo
        enddo
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do I=1,nd
ch           write(1,*)T(I),DUM(I),DF(I)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
        t(j) = azim(j)
dum(j) = vz(i,j)
enddo
        call smooth (nd,dt,fcc,dum,df)
do j = 1, npsi+1
        vz(i,j) = df(j)
        enddo
c....Plot
```

```
ch        open(1,file='try1.out',status='new')
ch          do II=1,nd
ch           write(1,*)T(II),DUM(II),DF(II)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End
        enddo

c_____________FORCE CONVERSION FOR WOPWOP_______________________

do 561 i=1,nrad-1
conv = 0.5 * rho  * (radius*omega*rad(i))**2.
conv1 = 48.22 * radius * rad(i)
c_________________________________
do 562 j=1,npsi+1
alfi = DTR*alpha(i,j)
c  lift and drag
c            dl=slift(i,j)
c            dd=sdrag(i,j)
            dl=thrust(i,j)
            dd=torque(i,j)

c  thrust and torque
c            FZ(I,J) = 4.*conv1*(dl*cos(-alfi)-dd*sin(-alfi))
c            FN(I,J) = 4.*conv1*(dl*sin(-alfi)+dd*cos(-alfi))

            FZ(I,J) = 4.*48.22*dl*radius*rad(i)
            FN(I,J) = 4.*48.22*dd*radius*rad(i)

 562    CONTINUE
c....Plot
ch        open(1,file='try1.out',status='new')
ch          do jj=1,nd
ch           write(1,*)azim(jj),fz(i,jj),fn(i,jj)
ch        end do
ch        close(1)
ch        call SYSTEM('gnuplot try1_plot.gpt')
ch        call SYSTEM('rm try1.out')
C....End
 561    CONTINUE
C
C ENSURE PERIODICITY
        DO I = 1, NRAD-1
        FZ(I,1) = FZ(I,NPSI+1)
        FN(I,1) = FN(I,NPSI+1)
        ENDDO
C_________________________________________________________________
C
c********** acoustic output ****************
if( acou ) then
write(12,3001) npsi+1,nrad-1
write(12,3002)(rad(i), i=1,nrad-1)
```

```
do 401 j=1,npsi+1
write(12,3003)(FZ(i,j),i=1,nrad-1)
 401 continue
do 402 j=1,npsi+1
write(12,3003)(FN(i,j),i=1,nrad-1)
 402 continue
endif
c********** Inflow output ****************
if( flow ) then
        write(13,*)'radial station = 0.1 to 0.3'
        write(13,5001)NRAD, NPSI+1
        write(13,5002)r1(1),r2(1),theta1(1),theta2(1)
write(13,3001) npsi+1
        do i = 1, npsi+1
        VzSum = 0.0
        do j = 1, 12
        VzSum = VzSum + VZ(j,i)
        enddo
        Vzz(i) = VzSum/12.0
        enddo
write(13,3003)(VZZ(i),i=1,npsi+1)
        write(14,*)'radial station = 0.1 to 0.3'
        do k = 1,npsi+1
write(14,3004)azim(k), VZZ(k)
        enddo
c+++++++++++++++++++++++++++++++++
        write(13,*)'radial station = 0.3 to 0.6'
        write(13,5001)NRAD, NPSI+1
        write(13,5002)r1(2),r2(2),theta1(2),theta2(2)
        do i = 1, npsi+1
        VzSum = 0.0
        do j = 8, 24
        VzSum = VzSum + VZ(j,i)
        enddo
        Vzz(i) = VzSum/16.0
        enddo
write(13,3003)(VZZ(i),i=1,npsi+1)
        write(14,*)'radial station = 0.3 to 0.6'
        do k = 1,npsi+1
write(14,3004)azim(k), VZZ(k)
        enddo
c++++++++++++++++++++++++++++++++++
        write(13,*)'radial station = 0.6 to 1.0'
        write(13,5001)NRAD, NPSI+1
        write(13,5002)r1(3),r2(3),theta1(3),theta2(3)
        do i = 1, npsi+1
        VzSum = 0.0
        do j = 20, nrad-1
        VzSum = VzSum + VZ(j,i)
        enddo
        Vzz(i) = VzSum/27.0
        enddo
write(13,3003)(VZZ(i),i=1,npsi+1)
        write(14,*)'radial station = 0.6 to 1.0'
        do k = 1,npsi+1
write(14,3004)azim(k), VZZ(k)
        enddo

        endif
C----------------------------------------------------------------
 1001    format(1x,'rotor # ', i2,/,'no of rad___no of azi'
     $        ,/,i3,7x,',',i3)
 1002 FORMAT(5x,'___radius___________r/R___')
 1003 FORMAT(3x,1Pe12.4,2x,1Pe12.4)
 1004 FORMAT(1X,'   AZI',' s cl    s cd',6X,
     >     'S LIFT',6X,'S DRAG',6X,'S LOAD',
     >     6x,'Thrust',5x,'Torq Fr',1x,'Alpha',2x,'Beta'
     >     ,' twist',3X,' vr','vtheta',3x,' vz',/,125('-'))
 1005 FORMAT(1X,f6.1,(' ',f5.2,'  ',f6.4),5(' ',e11.4),7(' ',f6.1))
 1007 FORMAT(1X,f6.1,(' ',f5.2,'  ',f6.4),5(' ',e11.4),7(' ',f5.1)
     $     ,1x,f5.2)
 1006 FORMAT (20A4)
c-----------------------------------------------------------------
 5001 format(1x,2i4)
 5002 format(2x,f10.4,2x,f10.4,2x,f10.4,2x,f10.4)
 3001 format(1x,2i4)
 3002 format(1x,5f7.4)
 3003 format(1x,5E16.8)
 3004 format(2(2x,E16.8))
 4001 format(1x,6f7.4)
 4002 format(1x,6f12.7)
        close(5)
        close(8)
        close(9)
        close(12)
        close(13)
stop
end
c
      subroutine SMOOTH(N,DT,FC,D,DF)
c Subroutine to smooth out artificial wiggles in the data caused by
c insufficient grid resolution. The routine uses FFT to filter out
c frequencies beyond a specified cutoff frequency.
c Arguments:
c N=number of points in array.
c DT=time interval between data points.
c FC=cutoff frequency, specified in units of 1/(time units).
c D=array of input data points.
c DF=output array of filtered points.
c Note that the work array WSAVE must be dimensioned at least 2*N+15.
c This routine must be compiled along with the FFTPACK library.
      implicit double precision (A-H,O-Z)
      dimension D(*),DF(*),WSAVE(1000)
      do I=1,N
        DF(I)=D(I)
      end do
      DELF=1.D0/((N-1)*DT)
      NFC=int(FC/DELF)+1
```

```
call RFFTI(N,WSAVE)
call RFFTF(N,DF,WSAVE)
NUP=N
if(mod(N,2) .gt. 0)NUP=N-1
do I=1+2*NFC,NUP
  DF(I)=0.D0
end do
call RFFTB(N,DF,WSAVE)
do I=1,N
  DF(I)=DF(I)/N
end do
return
end
```

F Hess Write Format

```
*     -*-FORTRAN-*-*
*     **************************************************************
*     *                                                            *
*     *  ********************************************************  *
*     *  *         Developed by Hormoz Tadghighi                *  *
*     *  *                                                      *  *
*     *  *             Boeing Helicopter Company                *  *
*     *  *                 Mesa,  Arizona                       *  *
*     *  *                                                      *  *
*     *  *       Output: Hess Format suitable for Surface       *  *
*     *  *               Meshing                                *  *
*     *  ********************************************************  *
*     *                                                            *
*     **************************************************************
*

*------------------------------------------------------------------------

C******** HESS FORMAT FILE
c*************************INPUT FILE FROM CADDY PROGRAM
      open(11,file='Blade.hess',status='unknown',form='formatted')
      DO 4002 K = 2,km
      DO 4002 J = 1,jm
      DO 4002 I = 16,im
      ICOLR = 0
      IFLG = 0
      IF (I .EQ. 16)THEN
      IFLG = 1
      IF (J .EQ. 1)IFLG=2
      ENDIF
      XH(I,J,K) = X(I,J,K)
      YH(I,J,K) = YYY(I,J,K)
      ZH(I,J,K) = Z(I,J,K)
      WRITE(11,1012)XH(I,J,K),YH(I,J,K),ZH(I,J,K),IFLG,ICOLR
 4002 CONTINUE
 1012 FORMAT (3F10.6, 2I1)
```

G 2D Gnuplot Plotting Routine

```
*     -*-FORTRAN-*-*
*     **************************************************************
*     *                                                            *
*     *  ********************************************************  *
*     *  *            Developed by Hormoz Tadghighi             *  *
*     *  *                                                      *  *
*     *  *             Boeing Helicopter Company                *  *
*     *  *                 Mesa,  Arizona                       *  *
*     *  *                                                      *  *
*     *  *       Output: Conversion of ROTTILT grid file        *  *
*     *  *               into Gnuplot format                    *  *
*     *  ********************************************************  *
*     *                                                            *
*     **************************************************************
*

*-----------------------------------------------------------------------
      program PLOTXY
      real XG(500),YG(500),XOBJ(500,20),YOBJ(500,20)
      real XGI(500),YGI(500)
      integer NPOBJ(20)
      character filnam*80,CH*3,XLBL*6,YLBL*6,TLBL*8,STR*80,BLANK*80
      do I=1,80
        BLANK(I:I)=' '
      end do
c Read grid point data file.
      write(6,*)' Enter name of grid file'
      read(5,'(a)')filnam
      open(1,file=filnam,status='old')
   5  continue
      read(1,'(a)')STR
      if(STR .eq. BLANK)goto 5
      read(STR,*)NXG,NYG
      do J=1,NYG
        YGI(J)=float(J)
        do I=1,NXG
          read(1,'(a)')STR
          if(STR .ne. BLANK)then
            read(STR,*)XG(I),YG(J)
            XGI(I)=float(I)
          end if
        end do
      end do
      close(1)
c Read object contour data and find min/max extent of object dimensions.
      write(6,*)' Enter name of object file'
      read(5,'(a)')filnam
      open(1,file=filnam,status='old')
      read(1,*)NOBJ
      XOBJMIN=1.E+30
      XOBJMAX=-1.E+30
      YOBJMIN=1.E+30
      YOBJMAX=-1.E+30
      do I=1,NOBJ
        read(1,*)NPOBJ(I)
        do J=1,NPOBJ(I)
          read(1,'(a)')STR
          if(STR .ne. BLANK)then
            read(STR,*)N,XOBJ(J,I),YOBJ(J,I)
            if(XOBJ(J,I) .gt. XOBJMAX)XOBJMAX=XOBJ(J,I)
            if(XOBJ(J,I) .lt. XOBJMIN)XOBJMIN=XOBJ(J,I)
            if(YOBJ(J,I) .gt. YOBJMAX)YOBJMAX=YOBJ(J,I)
            if(YOBJ(J,I) .lt. YOBJMIN)YOBJMIN=YOBJ(J,I)
          end if
        end do
      end do
      close(1)
      write(6,*)' Enter 1 for XY, 2 for YZ, 3 for XZ'
      read(5,*)IXYZ
      if(IXYZ .eq. 1)then
        XLBL='X AXIS'
        YLBL='Y AXIS'
        TLBL='XY PLANE'
      else if(IXYZ .eq. 2)then
        XLBL='Y AXIS'
        YLBL='Z AXIS'
        TLBL='YZ PLANE'
      else if(IXYZ .eq. 3)then
        XLBL='X AXIS'
        YLBL='Z AXIS'
        TLBL='XZ PLANE'
      end if
      XW1=XOBJMIN-0.2*(XOBJMAX-XOBJMIN)
      XW2=XOBJMAX+0.2*(XOBJMAX-XOBJMIN)
      YW1=YOBJMIN-0.2*(YOBJMAX-YOBJMIN)
      YW2=YOBJMAX+0.2*(YOBJMAX-YOBJMIN)
      if(XW1 .lt. XG(1))XW1=XG(1)
      if(XW2 .gt. XG(NXG))XW2=XG(NXG)
      if(YW1 .lt. YG(1))YW1=YG(1)
      if(YW2 .gt. YG(NYG))YW2=YG(NYG)
      IOUT=1
c Start plotting segment (IOUT=1 for X plot, 2 for PS plot).
  10  continue
      if(IOUT .eq. 1)then
        ISTAT=PGBEG(0,'/xwindow',1,1)
      else
        ISTAT=PGBEG(0,'/ps',1,1)
      end if
c Start new page.
      call PGPAGE
c Set viewport and window.
      call PGSVP(0.0,1.0,0.0,1.0)
      call PGQVP(1,VPX1,VPX2,VPY1,VPY2)
      D=min(VPX2-VPX1,VPY2-VPY1)/40
      VPX1=VPX1+8*D
      VPX2=VPX2-2*D
      VPY1=VPY1+8*D
      VPY2=VPY2-8*D
      call PGVSIZ(VPX1,VPX2,VPY1,VPY2)
      call PGWNAD(XW1,XW2,YW1,YW2)
c Draw and label frame around viewport.
      call PGBOX('bcn',0.0,0,'bcn',0.0,0)
      call PGLAB(XLBL,YLBL,TLBL)
c Draw horizontal grid lines.
      IY=0
      IY1=1
      IY2=NYG
      do I=1,NYG
        if(IY .eq. 1 .and. YG(I) .gt. YW2)then
          IY=0
          IY2=I-1
        end if
        if(YG(I) .ge. YW1 .and. YG(I) .le. YW2)then
          if(IY .eq. 0)then
            IY=1
            IY1=I
          end if
          call PGMOVE(XG(1),YG(I))
          call PGDRAW(XG(NXG),YG(I))
        end if
      end do
c Draw vertical grid lines.
      IX=0
      IX1=1
      IX2=NXG
      do I=1,NXG
        if(IX .eq. 1 .and. XG(I) .gt. XW2)then
          IX=0
          IX2=I-1
        end if
        if(XG(I) .ge. XW1 .and. XG(I) .le. XW2)then
          if(IX .eq. 0)then
            IX=1
            IX1=I
          end if
          call PGMOVE(XG(I),YG(1))
          call PGDRAW(XG(I),YG(NYG))
        end if
      end do
c Draw objects with interiors filled with background color.
      do I=1,NOBJ
        call PGSLS(1)
        call PGSLW(10)
        call PGLINE(NPOBJ(I),XOBJ(1,I),YOBJ(1,I))
        call PGSCI(0)
        call PGPOLY(NPOBJ(I),XOBJ(1,I),YOBJ(1,I))
        call PGSCI(1)
      end do
      call PGSLW(1)
```

```
c Reset viewport and window to maximum extent to permit drawing second set
c of axes.
      call PGQVP(0,XVMIN,XVMAX,YVMIN,YVMAX)
      call PGQWIN(XWMIN,XWMAX,YWMIN,YWMAX)
      BVX=(XWMAX-XWMIN)/(XVMAX-XVMIN)
      BVY=(YWMAX-YWMIN)/(YVMAX-YVMIN)
      AVX=XWMIN-BVX*XVMIN
      AVY=YWMIN-BVY*YVMIN
      XW1N=AVX
      XW2N=AVX+BVX
      YW1N=AVY
      YW2N=AVY+BVY
      call PGSVP(0.0,1.0,0.0,1.0)
      call PGSWIN(XW1N,XW2N,YW1N,YW2N)
      call PGSCH(0.75)
c Draw horizontal secondary axis.
      DEL=0.1*(YW2N-YW1N)
      call PGMOVE(XG(IX1),YW1-DEL)
      call PGDRAW(XG(IX2),YW1-DEL)
      do I=IX1,IX2
        call PGMOVE(XG(I),YW1-DEL)
        call PGDRAW(XG(I),YW1-0.875*DEL)
        if(MOD(I,20) .eq. 0)then
          if(I .lt. 10)then
            write(CH(1:1),'(i1)')I
            LCH=1
          else if(I .ge. 10.and. I .lt. 100)then
            write(CH(1:2),'(i2)')I
            LCH=2
          else if(I .ge. 100)then
            write(CH(1:3),'(i3)')I
            LCH=3
          end if
          call PGPTXT(XG(I),YW1-1.2*DEL,0.0,0.5,CH(1:LCH))
        end if
      end do
c Draw vertical secondary axis.
      DEL=0.08*(XW2N-XW1N)
      call PGMOVE(XW1-DEL,YG(IY1))
      call PGDRAW(XW1-DEL,YG(IY2))
      do I=IY1,IY2
        call PGMOVE(XW1-DEL,YG(I))
        call PGDRAW(XW1-0.9*DEL,YG(I))
        if(MOD(I,20) .eq. 0)then
          if(I .lt. 10)then
            write(CH(1:1),'(i1)')I
            LCH=1
          else if(I .ge. 10.and. I .lt. 100)then
            write(CH(1:2),'(i2)')I
            LCH=2
          else if(I .ge. 100)then
            write(CH(1:3),'(i3)')I
            LCH=3
          end if
```

```
    call PGPTXT(XW1-1.05*DEL,YG(I),90.0,0.5,CH(1:LCH))
  end if
end do
call PGEND
if(IOUT .eq. 1)then
  write(6,*)' Enter 0 to quit, 1 for Postscript file'
  read(5,*)I
  if(I .eq. 1)then
    IOUT=2
    goto 10
  end if
else
  write(6,*)' Postscript output is in file pgplot.ps'
end if
stop
end
```

H ROTTILT Input File for the XV15 Full Span Aircraft

```
******General Input File for the XV15 Full Span Aircraft***************

_DEBUG_ITEROP_ITERCL_LUNWR_LUNRS_LUNFL_LUNPL_LUNTB_LUNCT_
 false   , 200  ,    740   , 6    , 8   , 9   , 10   , 17,   18
_DETAIL___U______V___P_____W_____str Fun__GAM____RO__
TRUE, TRUE, TRUE , TRUE,  TRUE,  FALSE , FALSE, FALSE
__LROCON__LMUCON__LSOLID__LDATPR_____LUGRID__
  TRUE ,  TRUE   ,  TRUE ,  true     , TRUE
__LGEOMP__LUCOFP__LVCOFP__LWCOFP__LPCOFP__LTCOFP__LPLOTP__LFILDP__LPLT3D
FALSE    ,FALSE  ,FALSE  ,FALSE  ,FALSE  ,FALSE   ,TRUE , FALSE , TRUE
__MAX_ITERATIONS___ITEMOD____NU____NV____NW____NP____NT__
  850              , 10  ,   2   , 2  ,   2  ,  6   , 75
__RHO_(0.002377)__MU(3.719E-07)___RELAX-X__RELAX-Y__RELAX-Z_
    0.0023700  ,      3.719E-07   ,   0.1D0 , 0.1D0 , 0.1D0
_____UINF____VINF___WINF____PINF______TINF___REAL-GAS-CON__GAMMA
   0.0d0, 0.0D0, -1.0D0, 2116,        419.0, 1718.0D0,  1.4D0
__LINI__LOUTI__LINJ__LOUTJ__LINK__LOUTK__LIP___LJP___LKP__
  TRUE,  TRUE, TRUE,  TRUE,  TRUE,FALSE, TRUE, TRUE, TRUE
__IPREF__JPREF__KPREF___WINI__WOUTI__WINJ_ _WOUTJ__WINK__WOUTK
2     ,   64    ,  59   , FALSE, FALSE ,FALSE ,FALSE ,FALSE ,FALSE
__DRELXU__DRELXV__DRELXW__DRELXT__DRELXP__MODREL_RELMAX
   0.05  , 0.05  , 0.05 , 0.01  ,0.00   , 50  ,0.05
LC81__LSRSEC__LSMSEC__LCLCD___LCLCOR__LROTBOD
true,  false,   false,  TRUE,   true, FALSE
Velocity check__IUCK____JUCK____KWCK___XUCK___YVCK__ZWCK
                26,    22  ,    17  ,   0.0 , 0.0  , 0.0
_Number_of_Rotors_____DATA SET REFERENCE =14 NASA(TM X-952)__IBOUND
        2                                                  , 4
Name of airfoil used
MODIFIED VR SECTIONS
************>>>>>>>>> Data for rotor # 1 <<<<<<<<<<<**************
__Airfoil____# blades____Press dist(r/R)__
 1       ,      3.0D0,         0.250D0
Tip speed__Rotor radius_Hub radius(r/R)_Hinge Offse(r/R)_clock rot__
771.12D0,     12.5  ,   0.1353      ,   0.0357      , FALSE
Number Of Reference Rotor radius(r/R){max has been set to 15}f______
 44
Reference Rotor radius(r/R)________________________________________
.100, .125, .150, .175,
.200, .225, .240, .250, .260, .275, .290, .300, .310, .325,
.340, .350, .375, .400, .425, .475, .500, .525, .575, .600,
.625, .650, .675, .700, .750, .775, .800, .825, .850, .875,
.900, .915, .925, .940, .950, .960, .970, .980, .990, .995
CL cor(r/R)___Ref Twist__# Blade Div___Azi cor__# azi loc__conv fac__
   0.9D0,      -0.0D0,       100,     0.0D0  ,  720    , 0.083333333
# harmonic pit__# harmonic flap__FLAPPING__flap geo corr
2              ,    3     ,  FALSE        ,  FALSE
<<<<Harmonic_pitching_coefficients--Positive series-->>>>
 0,      8.1751D0,     0.000D0
 1,      0.0000D0,     0.000D0
```

```
<<<<Harmonic_flapping_coefficients--Positive series-->>>>
0       0.0000          0.0000
1       0.0000          0.0000
2       0.0000          0.0000
# data_points_______________________________________________________
21
nn_____r/R_____deflec_____Chord/RAD____CL des___T/chrd____Twist_____
 1,  .1353,    0.0D0,    .13780,      0.0E0,   0.12E0,   29.500
 2,  .2035,    0.0D0,  .13780,      0.0E0,   0.12E0,   25.500
 3,  .2700,    0.0D0,  .13780,      0.0E0,   0.12E0,   20.750
 4,  .3250,    0.0D0,  .13470,      0.0E0,   0.12E0,   18.625
 5,  .3750,    0.0D0,  .13470,      0.0E0,   0.12E0,   15.875
 6,  .4250,    0.0D0,  .13470,      0.0E0,   0.12E0,   13.500
 7,  .4750,    0.0D0,  .13470,      0.0E0,   0.12E0,   11.250
 8,  .5250,    0.0D0,  .13470,      0.0E0,   0.12E0,    9.000
 9,  .5750,    0.0D0,  .13470,      0.0E0,   0.12E0,    6.875
 10, .6250,    0.0D0,  .13470,      0.0E0,   0.12E0,    4.875
 11, .6750,    0.0D0,  .13370,      0.0E0,   0.12E0,    2.813
 12, .7250,    0.0D0,  .12330,      0.0E0,   0.12E0,    0.944
 13, .7750,    0.0D0,  .10990,      0.0E0,   0.12E0,   -0.833
 14, .8200,    0.0D0,  .10220,      0.0E0,   0.12E0,   -2.354
 15, .8600,    0.0D0,  .09710,      0.0E0,   0.12E0,   -3.771
 16, .9000,    0.0D0,  .09080,      0.0E0,   0.12E0,   -5.216
 17, .9350,    0.0D0,  .08330,      0.0E0,   0.12E0,   -6.500
 18, .9600,    0.0D0,  .07170,      0.0E0,   0.12E0,   -7.438
 19, .9750,    0.0D0,  .05730,      0.0E0,   0.12E0,   -8.000
 20, .9850,    0.0D0,  .04770,      0.0E0,   0.12E0,   -8.350
 21, .9950,    0.0D0,  .03810,      0.0E0,   0.12E0,   -8.700
************>>>>>>>>>> Data for rotor # 2 <<<<<<<<<<<<**************
__Airfoil____# blades____Press dist(r/R)__
 1       ,      3.0D0,         0.250D0
Tip speed__Rotor radius_Hub radius(r/R)_Hinge Offse(r/R)_clock rot__
771.12D0,     12.5  ,   0.1353      ,   0.0357      , TRUE
Number Of Reference Rotor radius(r/R){max has been set to 15}f_______
 44
Reference Rotor radius(r/R)_________________________________________
.100, .125, .150, .175,
.200, .225, .240, .250, .260, .275, .290, .300, .310, .325,
.340, .350, .375, .400, .425, .475, .500, .525, .575, .600,
.625, .650, .675, .700, .750, .775, .800, .825, .850, .875,
.900, .915, .925, .940, .950, .960, .970, .980, .990, .995
CL cor(r/R)___Ref Twist__# Blade Div___Azi cor__# azi loc__conv fac__
   0.9D0,       -0.0D0,       100,     0.0D0 ,  720    , 0.083333333
# harmonic pit__# harmonic flap__FLAPPING__flap geo corr
2              ,    3     ,  FALSE        ,  FALSE
<<<<Harmonic_pitching_coefficients--Positive series-->>>>
 0,      8.1751D0,     0.000D0
 1,      0.0000D0,     0.000D0
<<<<Harmonic_flapping_coefficients--Positive series-->>>>
0       0.0000          0.0000
1       0.0000          0.0000
2       0.0000          0.0000
# data_points_______________________________________________________
21
```

```
nn_____r/R_____deflec_____Chord/RAD____CL des___T/chrd____Twist_____
 1,  .1353,    0.0D0,    .13780,      0.0E0,   0.12E0,   29.500
 2,  .2035,    0.0D0,  .13780,      0.0E0,   0.12E0,   25.500
 3,  .2700,    0.0D0,  .13780,      0.0E0,   0.12E0,   20.750
 4,  .3250,    0.0D0,  .13470,      0.0E0,   0.12E0,   18.625
 5,  .3750,    0.0D0,  .13470,      0.0E0,   0.12E0,   15.875
 6,  .4250,    0.0D0,  .13470,      0.0E0,   0.12E0,   13.500
 7,  .4750,    0.0D0,  .13470,      0.0E0,   0.12E0,   11.250
 8,  .5250,    0.0D0,  .13470,      0.0E0,   0.12E0,    9.000
 9,  .5750,    0.0D0,  .13470,      0.0E0,   0.12E0,    6.875
 10, .6250,    0.0D0,  .13470,      0.0E0,   0.12E0,    4.875
 11, .6750,    0.0D0,  .13370,      0.0E0,   0.12E0,    2.813
 12, .7250,    0.0D0,  .12330,      0.0E0,   0.12E0,    0.944
 13, .7750,    0.0D0,  .10990,      0.0E0,   0.12E0,   -0.833
 14, .8200,    0.0D0,  .10220,      0.0E0,   0.12E0,   -2.354
 15, .8600,    0.0D0,  .09710,      0.0E0,   0.12E0,   -3.771
 16, .9000,    0.0D0,  .09080,      0.0E0,   0.12E0,   -5.216
 17, .9350,    0.0D0,  .08330,      0.0E0,   0.12E0,   -6.500
 18, .9600,    0.0D0,  .07170,      0.0E0,   0.12E0,   -7.438
 19, .9750,    0.0D0,  .05730,      0.0E0,   0.12E0,   -8.000
 20, .9850,    0.0D0,  .04770,      0.0E0,   0.12E0,   -8.350
 21, .9950,    0.0D0,  .03810,      0.0E0,   0.12E0,   -8.700
__no of rad stations for airfoil table_____ DATA IS DIFF for non c81__
6
__station at which airfoil table is considered___
0.00, 0.17, 0.3, 0.9, 0.95, 1.00
__logical unit for the airfoil tableS___
28
___The name of the file corresponding to lun = 28___
newcuff.c81
___The name of the file corresponding to lun = 28___
newcuff.c81
___The name of the file corresponding to lun = 28___
airtilt4.c81
___The name of the file corresponding to lun = 28___
airtilt4.c81
___The name of the file corresponding to lun = 28___
airtilt2.c81
___The name of the file corresponding to lun = 28___
airtilt3.c81
LWING__LNACEL__LV22BD
true,  true,  true
LTRIM__idtriml__ctreq__DELCOLM__TRIMTOL
true , 10  , 0.0107420d0 , 2d0  , 0.00001
```

I Hess Format Sample Data

```
*********** Sample Hess format file for the XV15 Fuselage***********
168   26
             X                   Y                  Z
       .11964832E+02      -.35352356E-08       .84840379E+01
       .12077237E+02      -.35352356E-08       .83474809E+01
       .12231729E+02      -.35352356E-08       .82237970E+01
       .12411149E+02      -.35352356E-08       .81239555E+01
       .12592197E+02      -.35352356E-08       .80379855E+01
       .12790360E+02      -.35352356E-08       .79569148E+01
       .12988524E+02      -.35352356E-08       .78974983E+01
       .13192248E+02      -.35352356E-08       .78379662E+01
       .13399462E+02      -.35352356E-08       .77829827E+01
       .13607160E+02      -.35352356E-08       .77362173E+01
       .13820728E+02      -.35352356E-08       .76899111E+01
       .14034295E+02      -.35352356E-08       .76466233E+01
       .14249661E+02      -.35352356E-08       .76060746E+01
       .14466899E+02      -.35352356E-08       .75664590E+01
       .14684138E+02      -.35352356E-08       .75301205E+01
       .14904112E+02      -.35352356E-08       .74962837E+01
       .15123381E+02      -.35352356E-08       .74632632E+01
       .15340245E+02      -.35352355E-08       .74329212E+01
       .15556381E+02      -.35352355E-08       .74050202E+01
       .15771334E+02      -.35352355E-08       .73803484E+01
       .15981738E+02      -.35352353E-08       .73672958E+01
       .16189506E+02      -.35352356E-08       .73691177E+01
       .16396501E+02      -.35352356E-08       .73709327E+01
       .16603693E+02      -.35352356E-08       .73727495E+01
       .16811307E+02      -.35352356E-08       .73745700E+01
       .17019323E+02      -.35352356E-08       .73763940E+01
       .17227781E+02      -.35352356E-08       .73782219E+01
       .17436751E+02      -.35352356E-08       .73800542E+01
       .17643641E+02      -.35352356E-08       .73818684E+01
       .17851419E+02      -.35352356E-08       .73836903E+01
       .18060514E+02      -.35352355E-08       .73855238E+01
       .18271396E+02      -.35352355E-08       .73873729E+01
       .18509707E+02      -.35352356E-08       .73894625E+01
       .18750294E+02      -.35352356E-08       .73915721E+01
       .18992769E+02      -.35352357E-08       .73936983E+01
       .19237920E+02      -.35352356E-08       .73958479E+01
       .20696137E+02      -.35352356E-08       .74086344E+01
       .20904720E+02      -.35352356E-08       .74104634E+01
       .21113045E+02      -.35352356E-08       .74122901E+01
       .21321398E+02      -.35352356E-08       .74141171E+01
       .21529758E+02      -.35352356E-08       .74159441E+01
       .21738120E+02      -.35352356E-08       .74177711E+01
       .22015920E+02      -.35352356E-08       .74202070E+01
       -------------      --------------        ------------
       -------------      --------------        ------------
```

REPORT DOCUMENTATION

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
January 1999

3. REPORT TYPE AND DATES COVERED
Contractor Report

4. TITLE AND SUBTITLE
A User's Manual For ROTTILT Solver: Tiltrotor Fountain Flow Field Prediction

5. FUNDING NUMBERS
C NAS1-20096
TA 7
WU 576-03-14-10

6. AUTHOR(S)
Hormoz Tadghighi and R. Ganesh Rajagopalan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Boeing Company
5000 E. McDowell Road
Mesa, Arizona 85215

8. PERFORMING ORGANIZATION REPORT NUMBER
L9KVJC-FR-98001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-2199

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA/CR-1999-208973

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Casey L. Burley; Tadghighi: The Boeing Company, Mesa AZ; Rajagopalan: Iowa State Univ., Computational Fluid Dynamics Center, Dept. of Aerospace Engineering, Ames, IA

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category: 71 Distribution: Nonstandard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A CFD solver has been developed to provide the time averaged details of the fountain flow typical for tiltrotor aircraft in hover. This Navier-Stokes solver, designated as ROTTILT, assumes the 3-D fountain flowfield to be steady and incompressible. The theoretical background is described in this manual. In order to enable the rotor trim solution in the presence of tiltrotor aircraft components such as wing, nacelle, and fuselage, the solver is coupled with a set of trim routines which are highly efficient in CPU and suitable for CFD analysis. The Cartesian grid technique utilized provides the user with a unique capability for insertion or elimination of any components of the bodies considered for a given tiltrotor aircraft configuration. The flowfield associated with either a semi or full-span configuration can be computed through user options in the ROTTILT input file. Full details associated with the numerical solution implemented in ROTTILT and assumptions are presented. A description of input surface mesh topology is provided in the appendices along with a listing of all preprocessor programs. Input variable definitions and default values are provided for the V22 aircraft. Limited predicted results using the coupled ROTTILT/WOPWOP program for the V22 in hover are made and compared with measurement. To visualize the V22 aircraft and predictions, a preprocessor graphics program GNU-PLOT3D was used. This program is described and example graphic results presented.

14. SUBJECT TERMS

Tiltrotor
ROTTILT
Fountain flow
XV15 Hover Acoustic Prediction
Rotor CFD Modelling
Rotorcraft Acoustic Prediction

15. NUMBER OF PAGES
88

16. PRICE CODE
A05

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL